Using the 3D shape of the nose for biometric authentication

Emambakhsh, Mehryar

Award date:
2014

Awarding institution:
University of Bath

Using the 3D shape of the nose for biometric authentication

submitted by

Mehryar Emambakhsh

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Electronic and Electrical Engineering

November 2014

Dedicated to my parents

Acknowledgements

I would like to express my gratitude to my supervisor, Dr Adrian Evans, for all of his kind support during my PhD career, including recommending my PhD application for the Graduate School Scholarship and financial supports, his brilliant comments and priceless constructive criticisms on my algorithms and suggestions to improve them, and proofreading my publications. I am also very grateful to the University of Bath Graduate School for sponsoring my PhD studies and the Department of Electronic and Electrical Engineering for providing the equipments and peaceful environment to encourage me further to pursue my research.

In addition, on behalf of our research group at the Department of Electronic and Electrical Engineering at the University of Bath, I would like to thank The National Institute of Standards and Technology (NIST) and University of Notre Dame for providing the access to the Face Recognition Grand Challenge (FRGC) dataset, Arman Savran (Bogazici University) for the Bosphorus dataset and University of Milano Bicocca for the UMB-DB 3D face dataset. I would also like to express my appreciation to the following researchers, whose publicly available codes were extensively helpful for the implementation of the proposed algorithms in this thesis:

- Vitomir Struc for his publicly available toolbox for face recognition, "The PhD face recognition toolbox", which was significantly helpful for my subspace projection algorithms implementation and ROC/CMC curves plotting,
- Vision Research Lab, University of California, Santa Barbara, for the implementation of Gabor wavelets codes,
- Professor Ajmal Saeed Mian, The University of Western Australia, for his bruteforce ICP implementation code,
- Jakob Wilm and Martin Kjer, Technical University of Denmark, for their K-d Tree-based ICP implementation,
- F. DÁlmeida, for the "Nonlinear Diffusion Toolbox",
- Daniel Claxton, for his "Surface Curvature" code, used in this work to find principal, mean and Gaussian curvatures and the shape index maps,

- Mehmet Öztürk for his nice and quick code for computing plane/surface intersection in discrete space.

Finally, a special thanks to my dad, my sister and brother-in-law, Khatereh and Ali, and brother, Mehrdad, for all of their kind support, especially, during the previous sad and difficult four years for the family. Last but not least, I would like to express my warm thanks to my best friends, Tony, Arash, Kapil, Rania, Mojca and Bala, for all the unforgettable moments and emotional support during the last years.

Mehryar Emambakhsh
University of Bath
November 2014

Abstract

This thesis is dedicated to exploring the potential of the 3D shape of the nasal region for face recognition. In comparison to other parts of the face, the nose has a number of distinctive features that make it attractive for recognition purposes. It is relatively stable over different facial expressions, easy to detect because of its salient convexity, and difficult to be intentionally cover up without attracting suspicion. In addition compared to other facial parts, such as forehead, chin, mouth and eyes, the nose is not vulnerable to unintentional occlusions caused by scarves or hair.

Prior to undertaking a thorough analysis of the discriminative features of the 3D nasal regions, an overview of denoising algorithms and their impact on the 3D face recognition algorithms is first provided. This analysis, which is one of the first to address this issue, evaluates the performance of 3D holistic algorithms when various denoising methods are applied. One important outcome of this evaluation is to determine the optimal denoising parameters in terms of the overall 3D face recognition performance. A novel algorithm is also proposed to learn the statistics of the noise generated by the 3D laser scanners and then simulate it over the face point clouds. Using this process, the denoising and 3D face recognition algorithms' robustness over various noise powers can be quantitatively evaluated.

A new algorithm is proposed to find the nose tip from various expressions and self-occluded samples. Furthermore, novel applications of the nose region to align the faces in 3D is provided through two pose correction methods. The algorithms are very consistent and robust against different expressions, partial and self-occlusions.

The nose's discriminative strength for 3D face recognition is analysed using two approaches. The first one creates its feature sets by applying nasal curves to the depth map. The second approach utilises a novel feature space, based on histograms of normal vectors to the response of the Gabor wavelets applied to the nasal region. To create the feature spaces, various triangular and spherical patches and nasal curves are employed, giving a very high class separability. A genetic algorithm (GA) based feature selector is then used to make the feature space more robust against facial expressions. The basis of both algorithms is a highly consistent and accurate nasal region landmarking, which is quantitatively evaluated and compared with previous work. The recognition ranks provide the highest identification performance ever reported for the 3D nasal region. The results are not only higher than the previous 3D nose recognition algorithms, but also better than or very close to recent results for whole 3D face recognition. The algorithms have been evaluated on three widely used 3D face datasets, FRGC, Bosphorus and UMB-DB.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human identification has traditionally played a crucial role in various aspects of human life. With the growth of the security concerns, the need of using unique features for recognising people has become more important. The necessity of having an authentic procedure to recognise authorised people from imposters has a wide range of application areas, such as in the military, industry sections and companies, universities, airports, border control, and even laptops, tablets and mobile phones. Without a reliable and robust analysis, an imposter can gain access to a system and can damage its security and integrity. The system can be a small orginisation with a limited number of employees or an airport with a very large throughput. In order to overcome this authentication issue, there is a need to accurately recognise people using a robust and reliable approach.

Costly and time-consuming manual recognition by human can be replaced by the automatic recognition approaches, performed by computer designed algorithms. Various methods have been used for this purpose. One of the most reliable approaches is to use the biometric features of subjects. For this purpose, physical or behavioural characteristics, which are unique for each subject, are detected, stored and compared with the subjects features held in a database. Alternatively, these features can be used to search through a dataset to try and find a specific identity.

The variety of biometric modalities available has resulted in numerous data acquisition and analysis algorithms. Fingerprints, irises and pupils, faces, palmprints, voice and gait are among the most widely used human biometrics. However, each of them has some disadvantages. The sensitivity of fingerprint and palmprint to scars and dirt on fingers, of iris recognition to contact lenses and blinking, of the face to make up, lighting conditions, pose and expression variations, and the time variance of voice and gait features have made robust biometrics a very challenging and difficult task.

Compared to the other biometric modalities, the face has many interesting characteristics.

Its everyday use for recognising by humans is well known. Also, the imaging devices for high quality face data acquisition are not very expensive. This is one of the reasons why there has been a huge amount of research in the field of face recognition over the previous decades. Figure 1-1 illustrates a block diagram of a simplified facial biometric imaging biometric system. The first step is data acquisition, in which the 2D colour and texture data of the subject's face are obtained in a biometric session. Then the data is post-processed and feature extraction/selection is performed. The gallery contains the biometric data obtained from the subjects in previous biometric sessions. The final feature vector from the probe image is compared with those obtained from the gallery samples and the subject's identity is eventually verified.

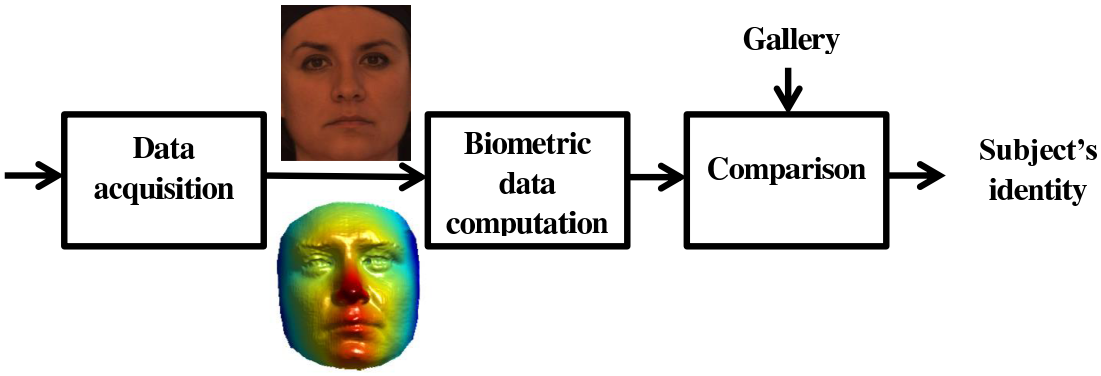
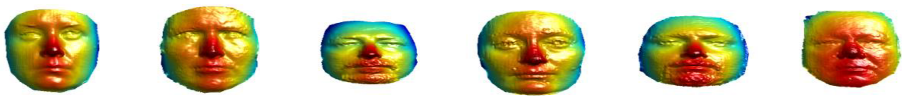


Figure 1-1: An abstract illustration of the 2D/3D face recognition algorithms.

In order to address the issues of 2D face captures caused by variations in lighting and pose 3D imaging has been introduced, enabled by the decrease of its cost and significant improvements in 3D image quality. 3D data capture also provides the ability to obtain facial curvature information and correct variations in pose, such that 3D face datasets have quickly become very popular for human identification.

Despite the use of 3D data, some of the initial face recognition problems are still unresolved and very challenging. These include variations in expression, occlusions, pose, and noise. In order to illustrate the significance of these problems on the performance of 3D face recognition, the iterative closest point (ICP) algorithm, which is a widely used approach for matching 3D models [1], is used to match a gallery consisting of six captures (Fig. 1-2-a), one from each of six subjects, with different probe images. The matching errors are then calculated as the value of the ICP's objective function after 25 iterations for each probe. For the first case, Fig. 1-2-b, the probe face is not occluded, has neutral expression, has the same pose as the gallery samples, and is not noisy. The lowest matching error is for the image first gallery sample, which is from the same subject as the probe sample and here is a correct recognition.

However, as shown in Fig. 1-2-c, when the same probe is partially occluded by a hand the ICP's matching error significantly increases, resulting in an incorrect match with gallery



1	2	3	4	5	6

(a) Gallery consisting of six captures, one each from subjects 1 to 6.

1.79	2.655	3.49	1.895	2.045	1.88

(b) Probe with neutral expression and frontal view and corresponding match errors.

10.66	13.22	**8.89**	13.85	9.74	13.425

(c) Probe with occlusion and corresponding match errors.

5.57	5.92	5.55	5.31	5.84	**5.24**

(d) Probe with different pose and corresponding match errors.

5.11	5.61	**4.91**	5.16	5.00	5.015

(e) Noisy probe and corresponding match errors.

2.53	2.63	2.92	2.24	2.945	**2.22**

(f) Probe with non-neutral expression and corresponding match errors.

Figure 1-2: Matching results using five different probe samples of subject 1 in the gallery with variations caused by occlusion, pose, noise and expression. The match errors with the gallery samples are found using the ICP algorithm [1].

subject. When the same probe image has different pose from gallery samples, the matching results also deteriorate. This is illustrated in Fig. 1-2-d, where the probe is rotated 45°around the yaw direction. The rotation causes partial data loss on the face, termed ”self-occlusion”, which will degrade the matching performance. In this case, instead of the correct match, the sixth gallery sample has the lowest matching error which is an incorrect match.

Noise is also an influential factor on the 3D face recognition algorithms. It can degrade the captured facial surface by adding random values to the point clouds. The noise effects are usually seen as spikes or holes over the 3D face, as shown in Fig. 1-2-e. Figure 1-2-e illustrates how ICP is vulnerable to noise in the probe sample, as the subject is wrongly classified as the

third gallery sample.

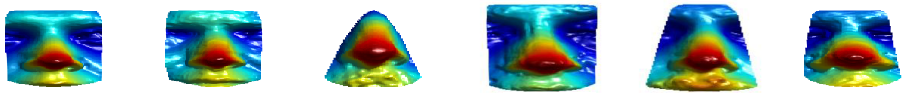
Facial expressions can significantly deform the face surface and as a consequence, decrease the performance of 3D recognition algorithms. An example of this case is plotted in Fig. 1-2-f, when the probe sample has a non-neutral facial expression an incorrect match with the sixth gallery subject. The above examples show how challenging the 3D face recognition task is, even for a gallery consisting of only six samples. In practice, the gallery and probe sets are much larger than this, which makes the recognition very difficult. A robust and reliable 3D face recognition approach needs to be capable of handling all of these problems, otherwise, the probe samples can be easily misclassified.

However, some parts of the face are relatively immune to these issues (except for noise) and a good example region is the nose. The nose's structure is relatively constant over different expressions and it is difficult to intentionally occlude it in biometric sessions. For example, compared to other parts of the face, the nose is very hardly ever occluded by scarves or hair.

In order to illustrates the potential of the nose for 3D face recognition, Fig. 1-3 shows the results of a similar evaluation to that in Fig. 1-2, except that the nasal regions from the captures are used. The nasal region is cropped by intersecting horizontal and vertical cylinders with the surfaces of faces, as described in [2], and then calculating the matching errors using the ICP algorithm. For the neutral, self-occluded and noisy samples shown in Fig. 1-3-b and -e, respectively, the correct gallery subject is recognised. Also, even for the occluded and non-neutral samples, see Fig. 1-3-c and -f, the matching errors of the correct subject (subject 1), are the second lowest of the errors found. This is a significant improvement on the results shown in Fig. 1-2 in which the whole facial region is used for matching. Although the same procedure, based on fixed distances, is used to crop the nasal regions for all samples in Fig. 1-3, the nasal region for gallery subject 3 is different from the other subjects because this subject has a larger face (and hence nose) compared to the rest of the gallery subjects.

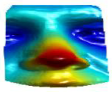
Considering the advantages of the nasal region, this thesis is dedicated to a thorough analysis of the potential of the 3D nose region for recognition. To date, there has been relatively little research specifically on addressing this problem. In addition, the preprocessing algorithms used for 3D faces are also analysed and a novel algorithm to correct the facial poses using the nasal region is proposed.

The application and effects of a number of widely used denoising algorithms for 3D face recognition are also evaluated with the aim of finding the best denoising method for holistic 3D face recogntiion algorithms on a noisy dataset. This approach also enables the effects of varying the denoising parameters to be evaluated and the optimal values, in terms of the overall performance of the face recognition system, found. In addition, an approach is proposed to learn the noise generated by the 3D laser scanners, which enables the simulation of noise with various power on any given 3D face data. Using this technique, the robustness of the



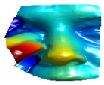
1	2	3	4	5	6

(a) Gallery consisting of six captures, one each from subjects 1 to 6.



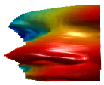
0.50	1.14	7.68	1.33	2.55	2.40

(b) Probe with neutral expression and frontal view and corresponding match errors.



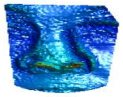
8.66	8.89	11.94	**8.53**	8.96	8.91

(c) Probe with occlusion and corresponding match errors.



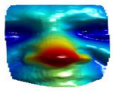
1.70	1.915	9.33	2.44	3.82	3.63

(d) Probe with different pose and corresponding match errors.



7.66	7.95	12.25	7.73	8.42	8.47

(e) Noisy probe and corresponding match errors.



1.92	**1.65**	7.62	2.00	3.46	2.79

(f) Probe with non-neutral expression and corresponding match errors.

Figure 1-3: Matching results using the nasal regions of five different probe samples of subject 1 in the gallery with variations caused by occlusion, pose, noise and expression. The match errors with the gallery samples are found using the ICP algorithm [1].

denoising and 3D face recognition algorithms can be quantitatively evaluated by changing the noise power. A novel algorithm is then proposed to detect the nose tip, which is one of the most basic landmarks on the face. The strength of the proposed approach is that it does not rely on predefined curvature thresholds and iteratively extracts candidate points for the nose tip. The algorithm is robust against occlusions and extreme facial expressions.

Two approaches are introduced to perform facial alignment over different expressions and in the presence of self- and partial occlusions. These are based on defining depth to binary map operators, which are able to consistently and accurately align the faces. Finally, the significance of the 3D shape of the nose for face recognition is shown by the definition of nasal

curves on the depth map and nasal curves and patches on the normals of Gabor wavelet filtered faces. As part of these algorithms, novel methods to localise the nasal region landmarks are introduced. The landmarks include more accurate localisation of the nose tip, nasal root, eye corners, alar groove and subnasale. A Genetic Algorithm (GA)-based feature selector finds subsets of the feature space which are more robust against expression variations. The results of the algorithm shows the highest recognition ranks ever reported on the 3D nose region, and are also higher than many previous 3D face recognition algorithms proposed in recent prestigious journal papers.

The algorithms have been tested on the widely used 3D face benchmarks, which are the face recognition grand challenge (FRGC), Bosphorus and UMB datasets. Figure 1-4 shows the structure of the remaining chapters of this thesis. While the first three chapters can be read independently, it is recommended that the fifth and seventh chapters are read before the sixth and eighth chapters, respectively.

The chapters are organised as follows. First, in chapter 2 an overview of the most recent and influential algorithms proposed in the literature is presented. In section 2.1.1, an introduction to biometrics and face recognition algorithms is provided while, in section 2.2, 3D face recognition approaches are explained in detail, including preprocessing (section 2.3.1), feature detection (section 2.3.2), feature extraction (section 2.3.3), a comparison of different approaches (section 2.3.4), post-processing methods (section 2.3.5), matching and classification algorithms (section 2.3.6), and decision making methods (section 2.3.7). A brief discussion on recent "deep learning" algorithms is provided in section 2.4. Existing work on the nose region for human identification is explained in section 2.5. An explanations on the widely used 3D face recognition datasets are provided in section 2.6 and the chapter is concluded in section 2.7.

In chapter 3, application of different denoising algorithms to 3D face recogntiion algorithms is described. After an introduction in section 3.1, the algorithm to model and simulate the noise is detailed in section 3.3, including the 3D face recognition pipeline used for the holistic approaches, which is described in section 3.3.1. The experimental results are then illustrated in section 3.4, including the classification algorithms in section 3.4.2 and denoising approaches in section 3.4.3. A comparison of the performance of denoising techniques is given in section 3.4.4 and the noise modelling and simulation results are provided in sections 3.4.5 and 3.4.6, respectively. Conclusions are given in section 3.5.

In chapter 4, the nose tip detector algorithm is explained. After providing a brief introduction to curvature, the basic weaknesses of using a fixed threshold are discussed in section 4.1. The new nasal region footprint is explained in section 4.2. The contribution made by the nasal footprint to the new nose tip detector is explained in section 4.3. The process of collecting a set of points as candidates for the nose's locations is explained in section 4.4 and, finally, the full

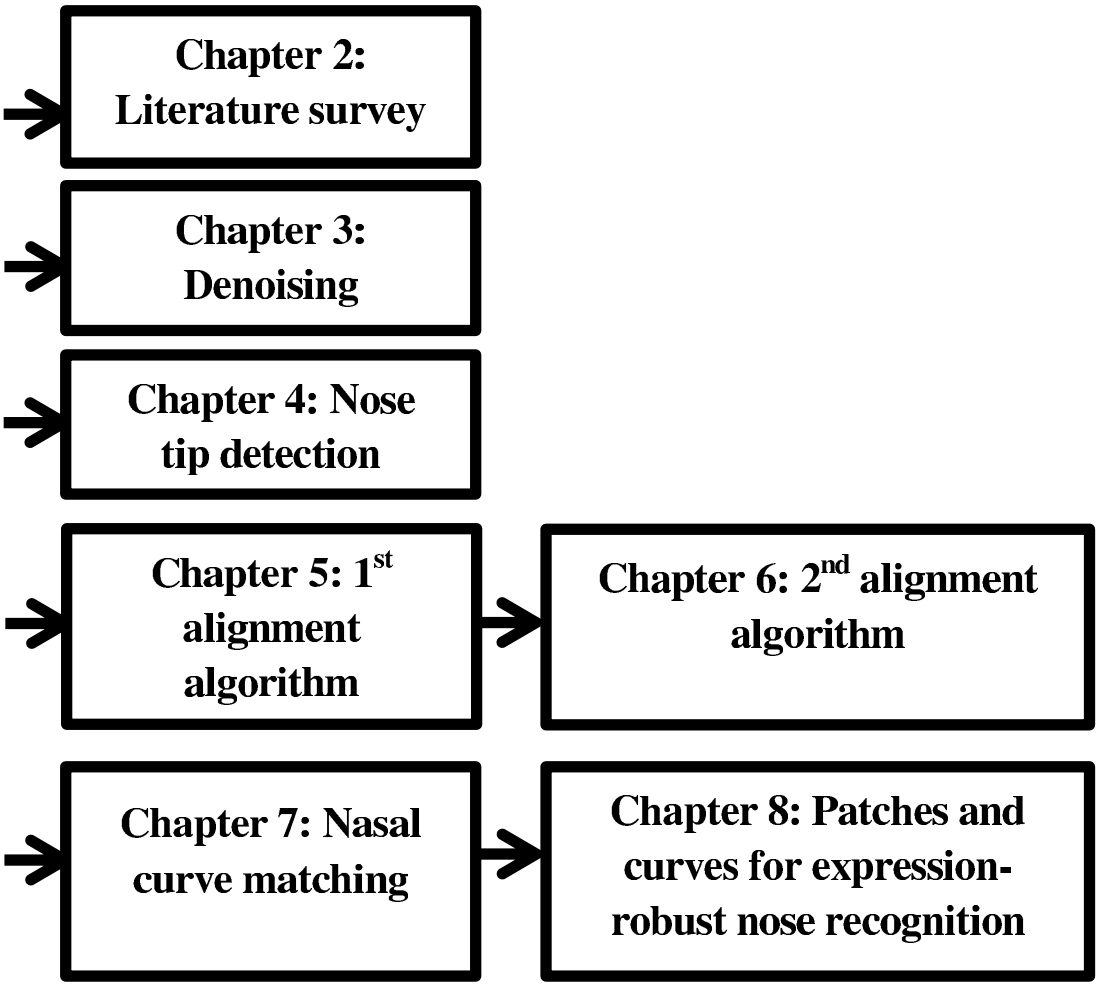


Figure 1-4: The organisation of the chapters and their recommended reading order.

nose tip detector algorithm is explained in section 4.5. Experimental results are provided in section 4.6, including an analysis of the variation of the algorithm's parameters and an accuracy analysis. The chapter is concluded in section 4.7.

Chapter 5 is dedicated to the initial facial alignment algorithm based on the nasal region. After a brief introduction on the importance of alignment for the 3D face recogntiion methods in section 5.1, the pose correction procedure is illustrated in section 5.2. The experimental results are then provided in section 5.3 and discussion and conclusions are given in section 5.4. Chapter 6 presents a significant improvement on the alignment approach in chapter 5, including a more thorough experimental evaluation on occluded and self-occluded samples. After an introduction to the PCA alignment algorithm (section 6.1), an improved approach using the nasal signatures is described in section 6.2. The alignment procedure is demonstrated in section 6.3, including an optimisation approach in section 6.3.4. Results are provided in sec-

tion 6.4, including samples with expression variations (section 6.4.1), quantitative within-class consistency (section 6.4.2), occluded samples (section 6.4.3) and robustness analysis (section 6.4.4). The algorithm's potential to be applied at lower resolutions is explained in section 6.4.5 and finally, the chapter is concluded in section 6.5.

Chapter 7 is dedicated to the nasal curve matching algorithm. First, a brief introduction is provided in section 7.1 and then the preprocessing algorithm is explained in sections 7.2, 7.2.1 and 7.2.2. The nasal landmarking approach and creation of the nasal curves are explained in sections 7.3 and 7.3.5, respectively. The feature selection process is detailed in section 7.4. Experimental results and conclusions are provided in sections 7.5 and 7.6, respectively.

The use of curves and patches on the nose over the normals responses of a multi-resolution Gabor wavelets filter bank is explained in chapter 8. In section 8.1, an introduction is provided, followed by the new nasal landmarking algorithm in section 8.2. A discussion of different feature types and the justification of using the Gabor wavelet response to normal vectors are given in section 8.3. Section 8.4 explains the feature descriptors and, finally, the GA-based feature selector and the matching criterion are described in section 8.5. Experimental results are provided in section 8.6, including the quantitative evaluation of landmarking algorithm's consistency and accuracy (section 8.6.1), the effects of varying the algorithm's parameters on the recognition performance (section 8.6.2), and an expression-robust 3D nose recognition evaluation (section 8.6.3). The chapter is finished in section 8.8.

The thesis concludes with chapter 9, which provides a summary of the proposed algorithms in section 9.1 and discussion and future work in section 9.2.

The main contributions of the thesis and its related field of science can be categorised as follows,

- An analysis of popular denoising algorithms for the 3D face recognition, including an evaluation of the sensitivity to their parameters are introduced for the first time.

- A novel method to learn and simulate the noise of 3D laser scanners is provided. This algorithm also shows those parts of the face which are less consistently reconstructed in the resulting 3D mode and enables a thorough quantitative robustness analysis of the denoising and 3D face recognition approaches.

- A robust algorithm, which is able to detect the nose tip and is not sensitive to facial expression, partial and self-occlusions is introduced. The approach is not vulnerable to the thresholding values of the curvature maps, which has been extensively used in the literature for the nose tip localisation.

- Two novel alignment techniques are introduced, which are based on new binary map operators. A thorough experimental analysis is performed on the partially occluded and

self-occluded samples. The consistency of the new algorithms has been quantitatively evaluated and compared with the widely used PCA algorithm.

- Novel techniques to extract the nasal landmarks are proposed. The algorithms do not rely on training data and have been tested for consistency and accuracy on two different datasets.

- Novel feature extraction and descriptors on the facial surface are proposed to recognise faces using the 3D shape of noses and surrounding regions. Unlike preciously reported algorithms, the new approach does not rely on sophisticated denoising and 3D registration algorithms. The outcome is a set of recognition ranks higher than any previously work, not only on the 3D nose recognition, but also higher or (very close to) the current state-of-the-art 3D face recognition algorithms.

Chapter 2

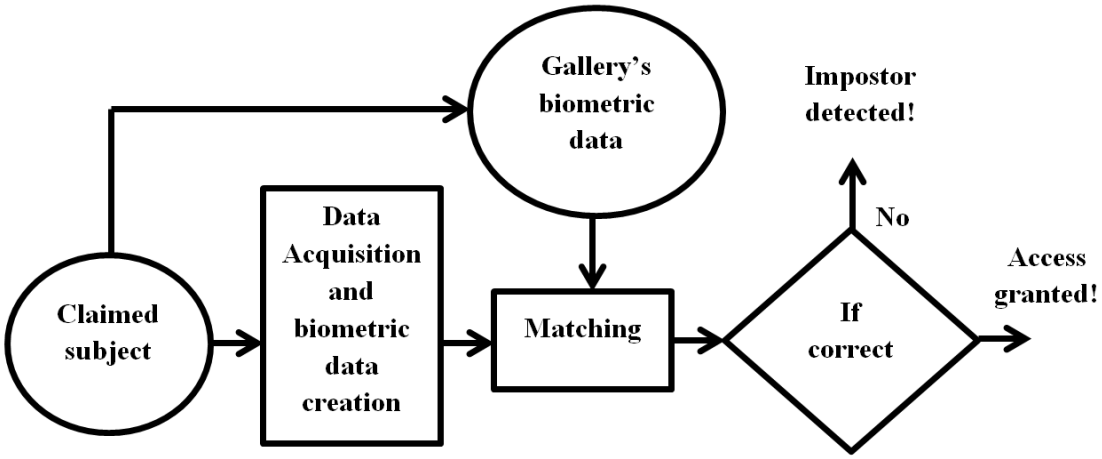
Literature survey

2.1 Introduction

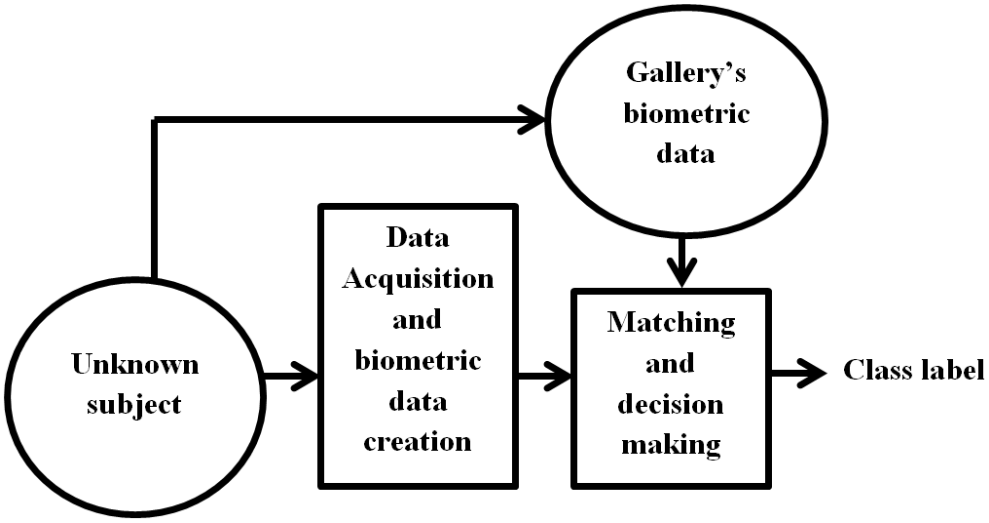
2.1.1 Biometrics

Biometrics is a combination of two Greek words [3]: bios, meaning life, and metrikos, meaning measure. It is the use of the science and technology to obtain behavioural or physical characteristics of a person to be used for identification or verification purposes. The sets of features, which have been extracted from a biometric modality are called the ”biometric data”. The main motivation of storing a person’s biometric data is to reuse it for the recognition purposes. Therefore, there are generally at least two sessions involved at a biometrics data storage process. The first one is called the ”data acquisition” stage, in which the raw data is obtained and biometric data is stored for the subject. At the second stage, the same data acquisition is performed, but this time the data is compared with the available dataset to check the person’s identity. The biometric data obtained at the first stage is usually known as the ”gallery”, while the one from the second stage is known as the ”probe”.

The person’s identity recognition is performed based on two scenarios. The first one is verification, in which the subject’s identity is compared with what has been claimed by him. This is shown in Fig. 2-1-a. The person’s biometrics data is obtained, and matched with the one in the gallery. If the matched score distance (or the similarity measurement) was higher than a threshold [4], the subject is approved. Otherwise, he would be rejected. The other scenario is called identification, in which the subject’s biometric data is compared with the whole dataset to find the best sample matching the subject’s biometric pattern. The lowest matching distance shows the correct subject’s label. Figure 2-1-b depicts different steps of this scenario. The matching procedure is clearly more time consuming than the verification scenario’s, because the comparison is performed between all gallery samples.



(a)



(b)

Figure 2-1: Two biometric scenarios: (a) Verification, in which the claimed identity of the subject is accepted or rejected by matching with its corresponding biometric data in the gallery; (b) Identification, in which the subjects identity is identified by matching its biometric data with all biometric data within the gallery.

2.1.2 Main steps in a biometric system

The block diagram shown in Fig. 2-2 can be used to explain different steps of a biometric system. The first step is preprocessing, which might includes denoising, resampling or alignment, data synchronisation (especially for multiple sensors) and filtering. The data can be obtained by various types of sensors or cameras. Like any electronic devices, these data acquisition tools produce noisy data. Noise is not a deterministic signal (process). Therefore, the noise on the gallery will highly likely be different from the noise in the probe data. As a consequence,

the matching and classification rate might be degraded. The denoising algorithm is used as an initial step to overcome this issue, as much as possible.

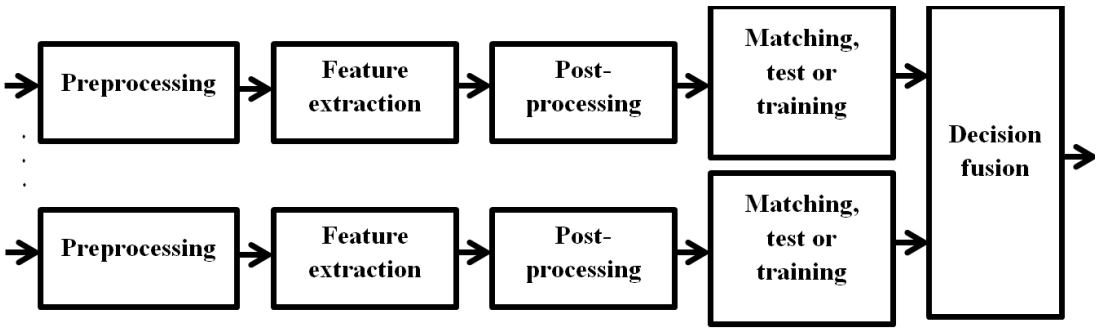


Figure 2-2: A rough approximation for a multi-modal biometrics system, whose final decision is made based on the ran-level fusion.

The second step is feature extraction. At this step, the feature space is created. The purpose is to quantitatively represent a biometric modality in a way to be different from other samples from other classes. Similar to any pattern recognition problem, there are two key aspects involved with the feature extraction algorithm. The first one is the within-class similarity (scatter) and the second one is the between-class dissimilarity. It is always intended to increase the feature vectors' similarity for the same samples, while the overlap of the features distribution to the other classes are reduced. This can be performed by a good feature extraction algorithm. On the other hand, the dimension of the feature space should be appropriately chosen. If the dimension is not large enough, the classes might not have enough separability. On the other hand, if the dimension is to high, there might not be enough number of samples in the feature space to be used to match with newer feature vectors (new dimensions). This highly important factor, which is known as *the curse of dimensionality* ([5]) should be avoided as much as possible. Otherwise, a high dimensional feature space might ironically produce lower classification rates than a feature space with a much lower dimension.

These issues sometimes require a post-processing algorithm to be used on the feature space. The feature space post-processing usually consists of feature selection, dimensionality reduction, mapping or feature vectors fusion. Finally, depending on the status of the system (either test or train phase), the post-process feature space is used to perform "matching" or "training" of a classifier. In the matching approach, the test pattern is compared with those in the gallery. The closest result is used to assign the class label. However, the training-based approach uses the samples in the gallery to train a classifier. A classifier is a function, which assigns a class label to an input feature vector. Matching-based methods are usually preferred. Because of the their flexibility for the change of gallery samples and also because they do not require a training stage. Finally, a fusion algorithm might be applied to the output of the decision mak-

ing step. The fusion is used to enhance the recognition performance, by merging the decisions especially found by a multi-sensor, multi-modal, or in general, when multiple criteria are used by more than one classifier/matcher.

A good biometric system should be [3, 4]:

- Consistent: The observed physical or behavioural biometrics features captured from a subject should not significantly change, when the probe data acquisition is performed. By ”significant change”, it is meant that the data should not lose its similarity with the data stored in the gallery for the same class labels, and should maintain its dissimilarity with the samples from other subjects. Since the change in the biometric data probe session is inevitable (such as the variations caused by aging or noise), the biometric system should be prepared to robustly handle such cases.

- Discriminative: The biometric data should not be similar between different people. Otherwise, it will lose its reliability to correctly recognise a subject.

- Easily obtainable: The imaging or in general, data acquisition procedure should be easy and practical for both the subject, whose biometric data is obtained and the biometrics system customer, who utilises the data capture device. Relying on sophisticated equipments might degrade the feasibility of a biometrics system.

- Robust: Another key feature of a good biometrics is its robustness. The biometric features should not be easily manipulated or changed by the subject. The biometrics system should be prepared to detect such change. Otherwise, the subject might be wrongly classified as another person or might not be correctly verified.

- Secure: As the acquired data is completely private for a subject, it should be very safely and securely stored. Also, the biometrics data should be properly coded and encrypted to avoid any spoofing attacks as much as possible.

- Maintainable: Although the storage costs are annually becoming lower and lower, the size of data should not be too large. This is not just to reduce the storage expenses, but to reduce the processing time of the data.

- Fast: The data acquisition and process procedures should be as quick as possible. Slow processing time is inconvenient for both sides of a biometrics system, i.e the subject and device customer.

As mentioned above, biometrics approaches are categorised into two types, the physical and behavioural [4, 6]. The physical biometrics are those which are assumed to be unique (or approximately unique) for each person. Some examples of these features that are mainly

obtained from the human anatomy are fingerprints, face, iris, pupil and palmprint. On the other hand, the behavioural features are those captured from an action performed by different subjects. A person's keystroke patterns [7] or patterns from daily activities, such as walking can be classified as behavioural biometrics [8, 9, 10]. Using the speech as a biometric approach has been also categorised as a combination of physical and behavioural feature [4].

To evaluate the performance of a biometric approach for verification purposes, four different criteria are usually considered [4]. The first one is true positive rate (TPR), which shows how much the approach can successfully classify the subjects to their claimed identities. The second one is the true negative rate (TNR), which refers to the rate of correct detection of an impostor. The two other criteria are the false positive rate (FPR) and false negative rate (FNR). FPR shows how the biometric system is vulnerable to detect an impostor as a legitimate person. On the other hand, FNR denotes the biometric system's imperfection in classifying a legitimate person as an impostor. A good biometric approach would have a high TPR and TNR, which in other words, means a low FPR and low FNR.

There are two widely used approaches to evaluate the performance of a biometrics system. The first one is by using the receiver operator characteristic (ROC) curve. In the case of a verification scenario (or open-set recognition [11], in which the subject might not necessarily have a corresponding sample in the dataset), based on the matching scores computed over the samples in a biometrics test session, a subject can be selected as legitimate or an impostor. The decision making is usually performed by assigning a threshold on the subject's biometrics matching scores [4]. All biometric systems produce some degree of errors in their output decisions. Therefore, if the decision making threshold is varied from 0 to 1 over a normalised set of matching scores, a subject's biometrics verification rate can be computed. Plotting these thresholds (false alarm rate (FAR) or false positive rate(FPR)) against the output verification rate (true positive rate (TPR)) or sometimes against FNR, produces the ROC curve. For the former, the more concave the ROC curve means a better verification rate. An important point on an ROC curve is when FAR and TPR are equal. This point is called equal error rate (EER) and is usually reported to show the biometrics system's verification performance. Also, detection error trade-off (DET) is a modified version of an ROC curve used to quantitatively describe the performance of a biometrics system. An example of an DET curve (whose vertical axis is simply $1-$ the ROC curve's) is plotted in Fig. 2-3.

The second approach is based on the cumulative matching characteristic (CMC) curve. The CMC curve is usually used for the identification scenarios (or in close-set recognition [11], in which the test subject has a corresponding sample in the gallery). It is the probability of correctly assigning the right class label to a test sample, when the matching scores are sorted per gallery samples [4]. Therefore, the classification rate at the i^{th} rank means the probability of assigning the correct label to the test samples, after the i smallest matching

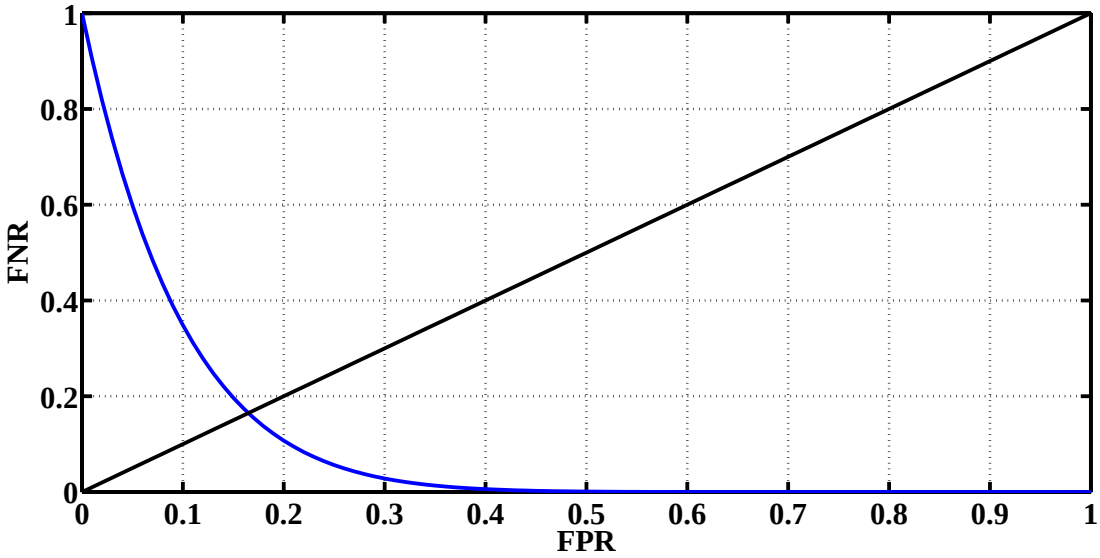


Figure 2-3: An example of a DET curve (blue) and the EER line (black).

scores are evaluated. For the CMC curves, usually the classification rate of the first rank is reported. It corresponds to the probability of the equality of test subject's label to the smallest matching score. An example of a CMC curve is depicted in Fig. 2-4.

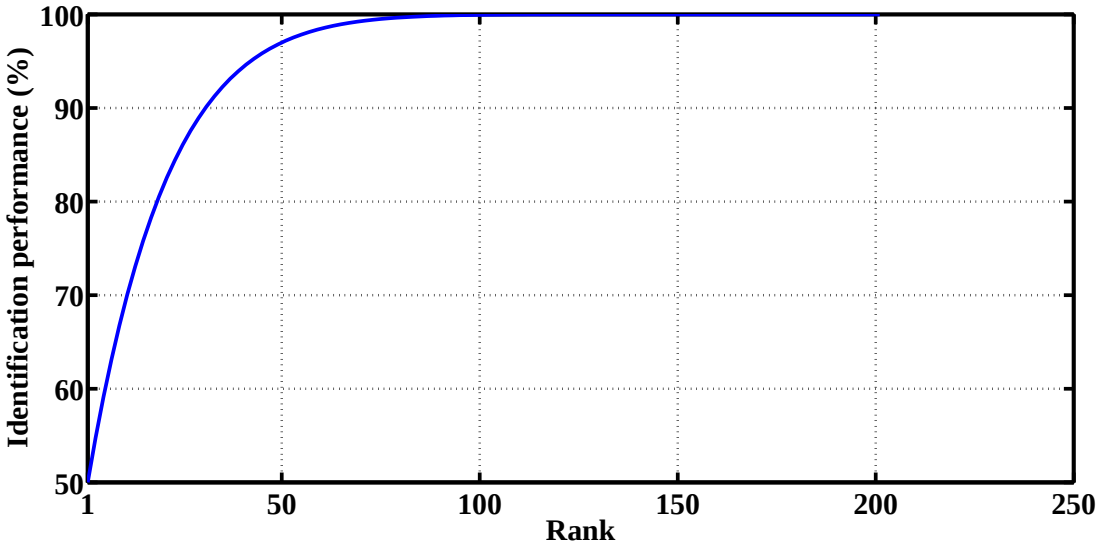


Figure 2-4: An example of a CMC curve.

In addition to the quantitative evaluations, there are some practical vulnerabilities involve with any biometrics. The iris recognition approaches, which use the iris texture for identifying a person are sensitive to contact lenses, segmentation, and blinking [4]. Also, fingerprint recognition can be degraded or sometimes becomes impossible when scratches or dirt are on

the finger [3]. The same issue can occur for palmprint recognition. Moreover, face recognition algorithms' performance can be deteriorated by make up, lighting variation, occlusion by scarves or hair, pose variation, aging and facial expressions [4]. The behavioural biometrics can also be sensitive to the data acquisition environment and based on the state of the subject might vary.

In their review paper, Jain *et al.* [4] introduces the latest advances in face, fingerprint, iris, voice and signature recognition. Multi-modal approaches, which use a combination of biometrics methods are also presented. Various methods for quantitatively evaluating a biometric algorithm is discussed as well. In [6], different machine learning and computational intelligence algorithms, such as neural networks and evolutionary computations for feature extraction and matching are explained.

2.2 Face recognition

The face has some interesting feature that has made it highly popular for human identification. In daily life, the face is used by humans to recognise others and, except for twins and other exceptional cases, its distinctiveness for each person is obvious. The imaging modalities used to collect the face datasets can be much cheaper than the other biometrics and the data acquisitions procedure is more straightforwards [12].

The quality of images captured by cameras has significantly improved during the last few decades. Meanwhile, the computational cost for image processing algorithms has also significantly decreased. The algorithms' implementation have become more straightforward and it has led to significant accomplishments in the object recognition research and technology. Face recognition, which is a branch of object recognition, has been the field of research during the previous decades. 2D face recognition refers to the detection and use of discriminative facial features obtained from the texture of two-dimensional gray scale or coloured facial images, for verification or identification purposes.

The fact that the face is highly discriminant between different people and the ease of data capture have made it very appropriate to be used for the biometrics systems. However, using the 2D facial data lacks some important features:

1. Sensitivity to lighting variations:
   The texture and colour appearance can be deformed when the lighting conditions vary in the biometric session. It can degrade the performance of those face detection (segmentation) algorithms, which are based on colour segmentation. In addition, the performance of the face recognition system highly relies on the within-class and between-class scatter, which might be degraded by lighting variations. Therefore, changes in the lighting

in between the training and test sessions can result in the subject's misclassification and decrease in the classification performance.

2. Sensitivity to pose variations:
   During the biometrics sessions, the subject might intentionally or unintentionally rotate his face. The rotation might decrease the feature detection accuracy and, as a consequence, degrades the face recognition performance. The pose can vary along the roll, pitch and yaw directions. If the rotation is only along the roll direction, it can be fixed using the 2D data. However, the 2D face information is not completely sufficient to correct the pose along the yaw and pitch directions.

In order to solve the aforementioned issues with the 2D imaging, 3D facial data acquisition was proposed during the last two decades. There have been significant improvements in the field of 3D imaging during the last decade. Laser sensors [13, 14, 15] and photometric stereo imaging [16] are the well-known approaches to collect the facial surface data and perform 3D reconstruction.

The resulting image is either stored as a mesh grid points or point clouds (point clouds are mostly used in the published datasets [13, 14, 15]). Due to their wide use in 3D face datasets, it is assumed that the 3D face is represented using the point clouds throughout this document. Utilising the 3D imaging for face recognition has numerous advantages, as outlined below:

1. Pose correction capability:
   Having the 3D facial data can help to correct the rotations along the yaw and pitch directions. The yaw and pitch pose variations can cause the loss of data during the data acquisition. This problem is known as "self-occlusion". However, if the rotations are not significant and the prior knowledge of the facial surface is used, the missing points can be approximated.

2. Lighting invariance:
   Lighting invariance is an interesting feature of the 3D facial imaging. It can provide the capability of more robust feature spaces, with higher separability than the corresponding 2D faces.

3. Provision of the facial curvature information:
   The most obvious application of the 3D imaging of the face is to have access to the facial surface information. The curvature can be calculated, which can be extremely helpful for feature detection and landmarking. Also, the curvature information can be used to segment the face and in many cases, outperforms the colour segmentation used by the 2D face detection/segmentation algorithms.

Face recognition has been the subject of considerable research over the recent years and there are many excellent survey papers on this topic. In [17] different recent algorithms for 3D face recognition are evaluated and compared. Moreover, in [12] various methods used for integrating 2D (colour or texture info) and 3D (depth info) are reviewed. A remarkable review paper about face recognition approaches is [18], in which different, mostly, 2D algorithms are thoroughly discussed. Furthermore, video-based face recognition algorithms, which use a sequence of images of different subjects' faces are also presented in this paper. A brief introduction about the psychological point-of-view for face recognition and visual attention used by humans to detect and recognise others' faces are also explained. Finally, different datasets and protocols used for evaluating a face recognition algorithm are illustrated. Another survey paper is [19], which introduces different method for 2D, 3D and hybrid face recognition algorithms. Other review papers include [20, 21, 22, 23, 24, 25].

2.3 Detailed overview of face recognition algorithms

A typical face recognition algorithm is plotted in Fig. 2-5. The main structure of the block diagram is very similar to the general biometric algorithms shown in Fig. 2-2. The preprocessing section of a face recognition algorithm usually consists of pose correction, denoising and face detection. Then, in the feature detection step, some informative sets of points, which can be consistently detected on the facial surface, are used for feature extraction. After that, the feature space is post-processed. Some of the well-known algorithms used for feature space post-processing are feature selection, mapping, dimensionality reduction algorithms. Multiple sets of features can be fused at this step to increase the classification accuracy.

The next step is denoted as feature space creation, in which the feature space can further be manipulated to make it more robust. There are numerous algorithms for this task, such as principal component or discriminant analysis algorithms. Then, the matching or classification methods are applied on the feature space and, finally, the decision making is performed. Depending the capabilities of a face recognition algorithm, some of these steps might be omitted. For instance, if the feature detection or extraction step is rotation invariant, using an alignment algorithm might not be necessary. In the following subsections, each of these steps are explained in detail. It should be mentioned that most of these algorithms cover a broad range of the previous research. Here the most popular, widely used and well-known approaches in the field of face recognition are explained.

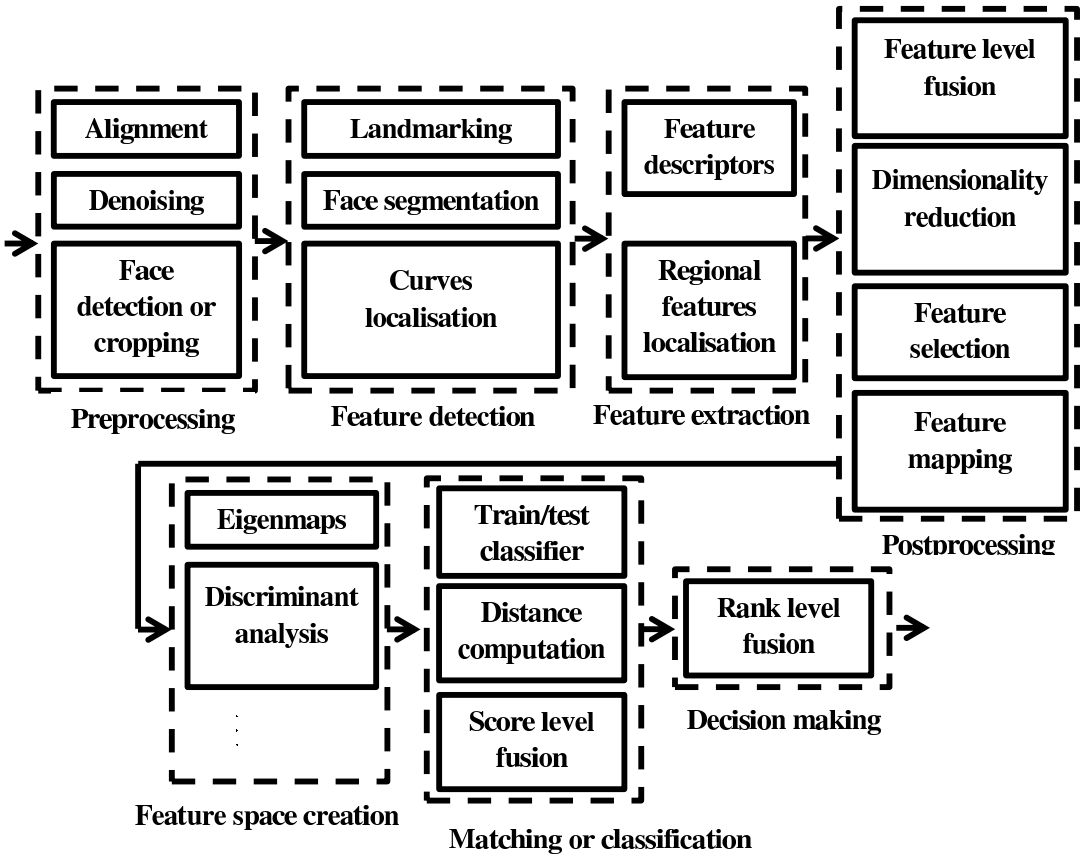


Figure 2-5: Different steps of a 3D face recognition algorithm.

2.3.1 Preprocessing

Denoising

The raw data obtained from the imaging device is usually preprocessed. The main reason for this is the presence of noise on the data. Part of the preprocessing is denoising and this can have a remarkable influence on the overall 3D face recognition performance. The widespread adoption of 3D laser scanners as the imaging modality for many 3D face datasets has led to a relatively consistent structure of denoising algorithms. Generally, four denoising steps are included: spike noise removal, surface smoothing, hole filling and missing data replacement. The holes and missing data differ in the sense that the depth values of the holes are available and are usually drastically lower than the neighbouring pixels. However, the locations with missing data do not have any values at all. Figure 2-6 shows examples of spikes, holes and missing data in a 3D facial image.

Spikes are a common noise produced by 3D laser scanners. As they usually cause impulsive variations on the face surface, median filters, the most popular approach for impulsive noise

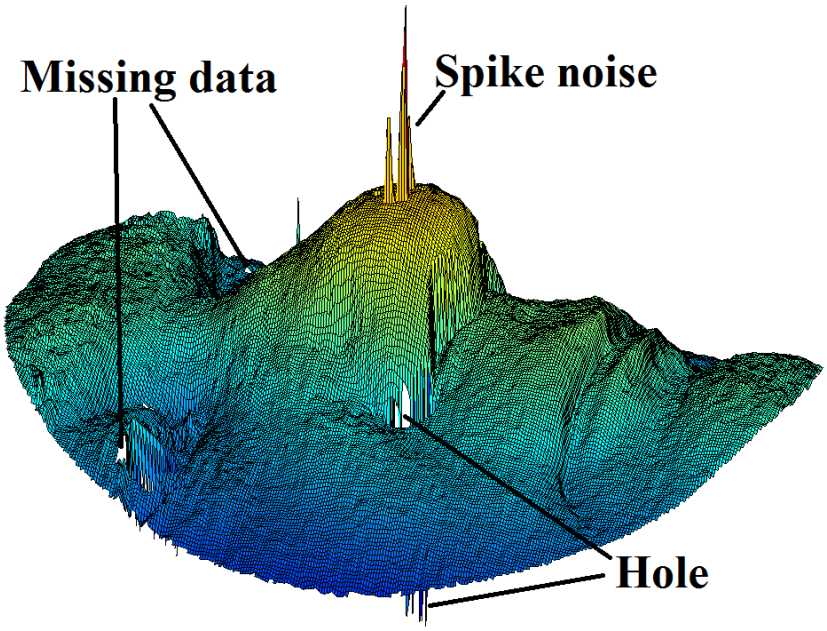


Figure 2-6: Different types of noise on a sample 3D face [26].

removal, are usually employed to remove them from the face's depth map, for example in [27, 28, 29, 30].

Spikes can also be interpreted as outliers in the data and, based on this assumption, regional statistical information of the pixels can be used to detect the noise locations [31]. In [32], the distances between the central point and its 8 neighbours are calculated. The standard deviation of the distances is then thresholded to detect the outliers. A similar approach is used in [33], except the neighbourhood is defined using an 11×11 mask and the angle between the optical axis and surface normal of the observed points is also used. The major issue with these algorithms is their sensitivity to the mask size and thresholds used. These parameters are usually set using trial and error on a given dataset but if the imaging modality changes retuning is required if over- or under-smoothing is to be avoided.

Surface smoothing is required as the raw depth image usually contains large variations caused by high-frequency components. These unwanted variations are not as salient as the spike noise and are spread over the surface. The most common approach to alleviate this type of noise is to apply low-pass filters on the facial surface. For instance, Gaussian [27, 28, 31, 34, 35, 29, 30, 36] and mean filters [37] are used to smooth and also remove the spikes from the depth surface. The median filter also has a smoothing effect and in [2] a 2.31×2.31 mm^2 median filter is applied to reduce the high-frequency noise effects. However, unlike the Gaussian filters, median filtering can move the image edges, which is not desirable. A different approach is to utilise the adaptive filtering to denoise the face surface, for example the Weiner

filter [38].

The main purpose of hole filling is to replace the holes and valleys on the face surface, which are produced by the inaccurate or low quality imaging. Morphological filling is one of the most popular algorithms for this task [39, 40]. Applying the filling algorithm directly on the depth map might unintentionally fill some natural holes on the face, such as the eye corners. In order to avoid this issue, the difference between the original and filled surfaces is calculated and those pixels with differences higher than a threshold are considered to be holes and are replaced using cubic interpolation [2].

Missing data is a common issue associated with the 3D imaging systems. It can occur due to self-occlusion (which appears after pose correction), large depth variations (for instance, due to open mouthes or the nostrils) or imaging device inaccuracy. These points are usually predefined as invalid points, which makes it easy for the dataset's user to detect them. Also, in some denoising approaches the noisy points are intentionally labeled as missing data and subsequently replaced [2, 32]. For all these cases, the most extensively used approach is to replace the missing data using interpolation. The main interpolation methods used are: cubic [32, 35], bicubic [31, 41], linear [33, 38, 37, 42] and K-nearest neighbours (KNN) interpolation [43].

Alignment

The other preprocessing operation that is usually applied on the 3D faces is the pose correction. If the feature extraction algorithm is sensitive to pose variations, the correspondence between feature vectors can be lost and the recognition performance drastically reduced. Therefore, a consistent alignment algorithm should be used to align the facial orientation to a predefined pose. There are numerous algorithms to perform this task, the most well-known being is the Iterative Closest Points (ICP) approach. 3D face alignment is the procedure used for removing posture variation from the images in the dataset and representing all of them in a unique pose. Depending on the degree of initial pose, it usually consists of two steps. The first one is the "coarse" alignment, which refers to a rough approximation of the post variation. The other step is the "fine" alignment, which is dedicated to locally detecting small variations from the optimal point.

In general, there are three different approaches to handle pose variations. The first of these is to utilise a rotationally invariant feature extraction method. The most common approaches are based on curvature [44] and, as such, can be sensitivity to outliers and noise, in addition to having a high computational cost. The second approach is to use a rotationally invariant representation for the points in the 3D space. Some examples are Extended Gaussian Images (EGI) [45], 3D Fourier descriptors [46], spin images [47], spherical harmonics [48, 49], and the snapshot algorithm [50]. The main disadvantages of these algorithms are their high

computational and storage costs. The third approach is to apply a pose correction as a preprocessing step, prior to the feature extraction, such that the faces are aligned in a predetermined pose [51]. Pose correction algorithms can broadly be classified as template based, landmark based or shape based (An excellent survey on face rotational alignment is provided by Murphy-Chutorian and Trivedi [51]).

Template based methods use a pre-aligned facial image for correcting the pose of the other images. The 3D test image is iteratively compared with a reference image chosen from the gallery or a general face model external to the dataset. Examples of this approach include the annotated face model (AFM) [28] and the use of adaptive active appearance models (AAM) [52].

Many methods have been proposed for the template matching task in the literature. Arguably the most important is the ICP algorithm [1]. ICP iteratively changes the rotation and translation of a 3D object (probe) in order to minimise an energy function defined by comparing the current state of the probe by the model (gallery) image. It has been extensively used for 3D face registration [53, 28, 54, 55, 33]. Although its initial algorithm was very computational expensive. Using the K-d tree [56], which is a binary search tree algorithm, to initialise the points makes ICP considerably faster. The other problem with ICP is that it can not detect big variations in pose [57]. In other words, it does not have a good performance in coarse alignments, with considerable pose variations from the optimal point. In these cases, its iterative algorithm can become trapped in local minima.

In a typical pose correction algorithm, landmarks are used for initialization and then ICP employed to register the facial surfaces to an average face model. An elegant extension is the two-pass approach of [58], which employs ICP in a second registration phase for individual regions. In [59], in order to solve the ICP's objective function sensitivity to local minima, a new measurement is introduced named Surface Interpenetration Measure (SIM). Silva *et al.* show that ICP's failure is because of the utilisation of the minimum squared error (MSE) as the cost function. Therefore, instead of MSE they use SIM with an improved GA optimisation algorithm.

The main problem of the template-based alignment algorithms is their sensitivity to the point-to-point correspondences in the error calculation. Although many different error functions have been proposed for 2D and 3D alignment, including the mean square error (MSE) [60], Normalised Cross Correlation (NCC) [61, 62], the Hausdorff distance [63, 64], and SIM [57], all point-to-point comparisons are very sensitive to occlusions such as those caused by scarves or hair. Impulsive spike noise in the depth data can have a similar effect and these issues can trap the alignment in local minima. Although ICP was originally proposed for 3D shape registration, it has had many applications for 3D face recognition. Another fast approach for calculating the ICP algorithm is to use the "Pre-computed Voxel Nearest Neighbor" [65].

This algorithm has been evaluated for both registration and recognition purposes on 3D ear images.

Landmark based methods rely on robustly detecting rotational invariant landmarks, which are then utilised for the pose correction. For example, in [66], curvature calculation, which is rotational invariant [44], is used to detect the landmarks. Different parts of the face are segmented by thresholding the mean and Gaussian curvatures and these parts are then used to correct the pose. The major drawback of this approach is the need for very accurate landmark detection, as even small errors in the landmark localisation can result in poor alignment. Localisation errors can be caused by noisy or missing depth data introduced during the image capture. Another problem that can occur during image acquisition is self-occlusion. This usually results from rotations around the yaw direction but can also be caused by rotations in the pitch direction. Self-occlusion can result in a substantial amount of missing data from one half of the face, causing the landmark detection algorithm to fail and leading to significant misalignment.

The final category of 3D pose correction techniques is shape based methods. These make use of prior knowledge of the face geometry and, unlike template based methods, do not require a reference image. Instead, either regional or holistic face information is used to remove the pose variations. In [36], the facial symmetry plane is iteratively detected by comparing each half of the face using the ICP algorithm and then used to correct the pose. However, as the face is not exactly symmetrical, especially over different expressions, this approach can fail to correct the pose accurately and consistently. In addition, asymmetric occlusions caused by hair, scarves or hands can trap the ICP algorithm in local minima. Another shape-based algorithm used for facial pose correction is Principal Component Analysis (PCA). PCA is widely used in pattern recognition to decorrelate feature vectors and for dimensionality reduction. Similarly, it is applied to 3D face data to map the points' distribution to the principal axes. For example, in the pose correction algorithm of [32], PCA is applied to the relatively well-aligned images in the FRGC v2.0 dataset.

Although PCA based face alignment is computationally efficient, is very appropriate for coarse alignment and can very successfully correct roll rotations, it has some noticeable disadvantages. Although it is reasonably robust to limited or symmetrical data loss, such as from self-occlusion due to pitch rotation, if the data loss is more significant it can fail completely. Problems can occur if some parts of the face are missing due to image acquisition problems, self-occlusion (in particular in the yaw direction), and deformations due to intense expression variation. In addition, when the 3D points are mapped to the principal axes the order of the axes can be changed and unwanted rotations and inversions introduced. These problems can be difficult to detect and require a post-processing stage in order to correct them.

Face detection and segmentation

The location of the face is not necessarily predetermined in the face dataset and in real outdoor applications. Therefore, in the initial steps of a face recognition algorithm, there should be a face detector to localise or crop the face. Numerous algorithms have been used for face detection and segmentation. The most common detection approaches rely on colour and curvature information. The colour space is usually transformed into YCbCr [67, 40] and, since the human skin colour has a specific range of variation in this colour domain [67], a simple thresholding algorithm can be successfully applied for skin detection. After that, with some post-processing algorithm and using the prior knowledge about the face geometry, the face is localised more accurately. More sophisticated face detection and segmentation methods can be found in the literature for this task, such as using neural network, Support Vector Machines (SVM) classification, Bayesian and Gaussian Mixture Model (GMM) clustering algorithm.

Another approach which is becoming more popular for face detection and segmentation, is the use of depth information only. Instead of using the colour information, the Gaussian and mean curvature ($\mathbf{H}$ and $\mathbf{K}$, respectively) [66] are used to perform the segmentation. In this type of face detection algorithm, curvature calculation (more specifically, Gaussian, mean, principal curvatures or the Shape Index (SI)) is utilised [66, 63]. Since curvature calculation is rotation invariant [68], prior knowledge of the face surface can be employed to find the specific thresholds for different parts of the face. For example, a common way to locate the nose tip, which is one of the most important landmarks on the face surface, is to find the biggest connected convex region. This is usually achieved by applying some thresholds to the Gaussian and mean curvatures.

The nose tip's location is often used for 3D face cropping. Typically, a sphere is centralised on the nose tip and intersected with the facial points. The inner parts of the sphere are used as the face. The radius of the sphere is chosen using the $\mathbf{X}$ and $\mathbf{Y}$ (resolution maps), found from the point clouds. If the resolution maps are not available (for example in a photometric stereo imaging [16]), the face cropping will be more complicated and the methods, which utilise the facial shape prior information should be used.

2.3.2 Feature detection

There are some points on the facial region, which can be (relatively) consistently detected. These points are known as facial landmarks or keypoints [69, 70, 30, 63, 44, 68, 71]. In addition to the landmarking algorithms which use the prior shape knowledge of the face, there are other automatic landmarking methods based on gradient operators. These approaches, which are affine transformation (scale, rotation, and translation) invariant, and are robust to view-point and intensity variations, can be classified into corner detectors and Scale Invariant

Feature Transform (SIFT) operators [72]. The edges of a 2D image (or a depth map) include an abundant amount of information, which has also been shown to be used by human eyes for object detection and recognition.The basis of the corner detector algorithms is to evaluate the pixel-wise or regional gradient of an image. To be more specific, the 2×2 covariance matrix of the pixels' gradient in a given mask is computed. The two eigenvalues of the covariance matrix are then calculated and used to determine whether a pixel represents a corner, an edge or a uniform region.

For example, in Kanade Lucas Tomasi's corner detector ([73]), a point denotes a corner if the minimum of its eigenvalues is higher than a given threshold. Iteratively, an edge occurs when one eigenvalue is significantly higher than the other. A constant (uniform) region usually has equally small eigenvalues. Harris corner detector [74] uses a Gaussian kernel to smooth the gradient image and then applies a similar procedure to find the eigenvalues. However, the comparison criteria is different and a nonlinear function is applied over the eigenvalues to create a "cornerness" measurement. The resulting map is then thresholded to detect the corners and edges. Both of these methods are rotation invariant. This is because the eigenvalue computation from the covariance matrix is a rotation invariant method. Also, regardless the direction of the eigenvectors, the eigenvalues are treated equally, which again helps the rotation invariance feature.

However, these methods are scale variant. If the viewpoint changes in a way which results in a larger or smaller object, the algorithms might fail to choose the right edges and improve this feature, the operators have to be applied over multiple scales. The input image is filtered (to minimise the high frequency overlap) and downsampled to produce a pyramid of images. The corner detectors are then applied on the pyramid's images. This leads to the use of SIFT operators [75]. The SIFT algorithm consists of three steps: interest point detection, description and matching. Here the keypoint detection step (the first part) is explained. First, difference of Gaussian (DoG) maps are computed over the the input image's different scales. Then, for each scale, extrema detection is applied. To do this, the Taylor expansion of the DoG map is approximated and its extrema by differentiating and setting to zero. The location of the extrema is then applied to the DoG. The weak edges, which have insignificant absolute DoG values and are along the keypoints, are removed. A quicker implementation of the SIFT algorithm is Speed-Up Robust Features (SURF) ([76, 77]), which uses approximated binary masks to compute the DoG.

2.3.3 Feature extraction

Feature extraction is the core of a face recognition algorithm. In the literature, numerous approaches have been proposed to perform this task and some of the most important ones are direct used of depth maps, point clouds, multi-resolution wavelets [36], LBP [78], DCT [79],

EGI [45, 80], spin images [47], Gaussian and mean curvature [66, 81, 80, 82, 68, 83], Gabor filters [37, 84] and normal maps [38, 85]. Based on the type of further segmentation applied on the facial region, the feature extraction methods might differ. In general, the face recognition algorithms can be roughly categorised into three sub-classes [18]: 1) holistic methods; 2) regional algorithms; 3) landmark and curve-based methods. Algorithms which use the whole facial region for feature extraction, are known as the "holistic" face recognition approaches. The use of the wavelet transform for setting up feature space is explained in [28]. After aligning the model in three steps, using spin images [47], ICP and SA on the differences between the depth information of the gallery and probe, Haar wavelet and pyramid transforms are used for feature extraction.

Similar to [32], in [54] both texture and depth information, captured by photometric stereo imaging, are used for recognition. After registration using ICP, a similarity criterion is defined for the gallery and probe images. Unlike other approaches which only took into account the depth info in their similarity measurements, here texture information is also considered. The algorithm has been evaluated on a dataset including 62 subjects with different postures.

Another holistic approach introduced a method based on calculating the differences between Haar, Gabor and Local Binary Patterns (LBP) for gallery and probe are calculated [36]. In order to make their approach robust for expression variations, Binghamton University 3D facial expression (BU-3DFE) dataset and a boosting classifier are utilised to detect the expression before face recognition.

Another example of a holistic algorithm is presented in [86]. In this approach, computer graphics techniques are utilized in order to evolve an average 3D textured face model. To be more specific, PCA is used to define a changeable 3D face model from a dataset of 3D images. Next, the average shape model is evolved to be fitted to the gallery images. After convergence, this procedure results in a vector of fitting parameters. In the test phase, the same algorithm is used to calculate the fitting parameters for the probe image. These parameters are then compared for the probe and gallery images in the matching step. Mapping the 3D face surface to a 2D image has also been used [87]. In their algorithm, after detecting the nose tip and cropping the face region, the depth values are isomorphically mapped to two dimensions. Finally, the eigenface algorithm is employed on the mapped images for recognition.

In [38], instead of using the depth or points' coordinates for 3D registration, the point's normals are used in conjunction with a Fisher's discriminant paradigm. In other words, the points' normals , which maximise the concentration of within-class scatter, while simultaneously maximize the between-class distribution, are selected for an expression invariant approach. The result is one of the highest verification rates ever reported on FRGC ROC III experiment, which is 99.2% at 0.1% FAR. In a quite recent paper [85], facial normals' histograms, captured from multiple rectangular regions are used to set up an expression-robust feature space and a novel

sparse classifier is defined to perform the face matching step. As a result, 96.3%, 98% and 94.2% rank-one recognition rates are obtained for FRGC v2.0 using neutral and non-neutral probes, respectively.

Deformations caused by facial expression are downgraded in [36] using a method, named as shape difference boosting. This approach uses the Bosphorus dataset to learn the expressions and signify those facial regions which remain constant at different expressions. Then, these constant parts are boosted on the FRGC v2.0 dataset, resulting in 98.39% rank-one recognition rate, while 98.04% at 0.1% FAR is obtained or ROC III experiment.

Another approach for 3D face recognition uses regional information extracted from different parts of the face. The features vectors from these regions are then combined. The segments are either detected using facial segmentation [67, 88, 58, 57] or extracted using their performance for an expression invariant face recognition [38, 36, 89, 85]. In [88], a multiple regional approach is used based on a PCA-LDA feature extraction method. The regions recognition scores are fused to finalise the decision making. The result provides one of the highest recognition ranks are obtained on the FRGC v2.0 dataset, which is 99.0%. Also, the result of ROC III is 94.6% for 0.1% FAR. One of the regions fused by this algorithm is the nasal region. A 94.5% rank-one rate and 83.7% 0.1% FAR are obtained.

In [90], instead of considering the whole range data, first mean and Gaussian curvature maps are calculated in order to extract the convex regions. Then, EGI, which are rotation and scale invariant, are calculated for these regions. Finally, graph matching is used for recognition.

Similarly, in [80], EGI is computed on convex and concave points, found by curvature calculation on two different faces. Then, spherical correlation coefficients [91] are extracted and used for matching. A similar approach is used in [82]. After curvature calculation on the face surface, the Hausdorff distance is utilised for recognition. Also a weighting is used for the distance calculation according to the depth information. The weighting is performed using the curvature maps, which consist of principal, Gaussian and mean curvatures. The points with bigger curvature are weighted higher.

Another regional algorithm has been proposed in [81]. Similarly, curvature calculation is used to segment the face. Then eight sets of features are found from these segments: eyes, nose and head width; eyes span; nose height and depth; eye separation; Gaussian curvature, maximum Gaussian curvature and average minimum curvature on the nose ridge, bridge and base. The algorithm is tested on a small dataset including 24 subjects.

Four different regions (entire face, circular and elliptical nose area and upper face) are combined in [57]. Then M-estimator sample consensus (MSAC) is used for coarse registration. Also, a measure named SIM [59] is employed for fine registration. This measurement produces some matching scores which are used for recognition. Alyuz *et al.* used seven non-overlapping regions on the face surface for recognition [58]. A method based on average face

model (AvFM) is used for registration. Curvature calculation is used for landmark detection and face segmentation. The feature space is set up using three sets of features. The point cloud features, statistical point cloud feature and curvature based 3D shape descriptor. Finally, different types of fusion methods are tested on the resulting feature vectors.

After curvature calculations and thresholding, the face surface is segmented into different regions in [92]. About 86 feature sets are extracted from the regions and after feature selection, these features are combined to use for the recognition. The algorithm has been evaluated on a dataset including 420 images of 60 subjects. Chang *et al.* proposed another regional 3D face recognition algorithms [33]. Colour information and curvature calculation are used for skin detection and face segmentation. Then, a number of different regions centred on the nose are extracted. For the test phase, the same procedure is performed on the probe image and ICP is ustilised to iteratively find the minimum registration error between the two images. The algorithm is compared with the holistic ICP and eigenfaces. The results show the superiority of the regional matching over holistic approaches when considering expression variations.

A regional registration algorithm in conjunction with LDA classifiers are used by Alyuz *et al.* for expression robust 3D face recognition [58]. The results on FRGC v2.0 are 97.5%, with a 1.91% EER on ROC III experiment. In their approach, it was observed that that the nasal region has high discrimatory power, as evidenced by its 91.81% rank-one recognition rate. Another research focusing on multiple regions integration for 3D face recognition is provided by Queirolo *et al.* [57]. Four different regions are segmented and stored for the gallery sessions, two nasal regions, an upper face image, and the whole face. The regions are then matched using a novel matching criterion, termed as the SIM and SA. This approach achieves one of the highest rank-one recognition rates, 99.6% for FRGC v2.0.

Another approach for face recognition is based on using a set of landmarks on the face region in order to set up the feature space. These landmarks are located on the face surface and their relative geometrical relationship is usually calculated. To overcome the holistic face recognition algorithms sensitivity to expression variations, Mian *et al.* proposed a landmark-based method, joint with a localised feature descriptor, to incorporate the 2D texture and 3D point clouds for an expression-robust face recognition. The rank-one recognition rate for the 3D face recognition on FRGC v2.0 is 93.5%. A 99% rank-one recognition rate is achieved for the neutral probe set. However, it decreases to 86.7% when non-neutral samples are used.

Alternatively, the landmarks on the face surface can be used to perform recognition using features introduced by some contours or lines intersecting the landmarks. In [93], the intersection of three planes with the face surface are found. The recognition step is performed by comparing the L_2 norm of the gallery and probe images intersections. The planes are yz (vertical), xz (horizontal) and a cylinder whose center is located on the nose tip. Various locations are chosen for the horizontal and vertical planes with $1mm$ resolution. Also, different radii for

the cylindrical plane are intersected.

A completely different approach for face recognition is based on using concentric contours on the face surfaces [94, 95]. In [94], after setting up these contours using the fast marching method [96], they are mapped to a canonical form (isometric Riemannian surfaces). Then, in the test phase, these distances are compared for the probe and gallery images. Although setting up these geodesic contours is very computational expensive, the algorithm shows robustness for expression variations.

Another commonly used algorithm for 3D face recognition is using the facial curves, which can also be categorised as regional algorithm. A set of connected points on the surface of facial region create a curve. The curves are very good descriptors of the face surface. Unlike landmarks, they are not too localised, they have lower dimension than the regional methods and have shown robustness against expression variations [97, 98, 99, 100] and occlusion [42].

In [42], radial curves computed by intersecting planes with the face, which pass through the nose tip. Then their quality is assessed in order to handle missing data and occlusions. Using a very low dimensional selected curves, they obtained 97.0% recognition rate on FRGC v2.0 dataset. Also, 99.2% and 96.8% rank-one recognition rates are found for the neutral and non-neutral probes, respectively. Another curve-based algorithm is proposed by [97]. First keypoints are detected on the facial surface, then the least variant curves on the face are selected using a statistical model and matched with those in the gallery.

This approach results in 95.6%, 97.3%, and 92.8% rank-one recognition rates for FRGC v2.0, neutral and non-neutral probes, respectively. As an extension over the curves, "Iso-geodesic Stripes", centralised at the nose tip, are used in [99] for an expression invariant 3D face recognition. A novel descriptor named "3D Weighted Walkthroughs" is utilised to quantify the differences among the corresponding stripes. The recognition ranks obtained on the FRGC dataset are 94.15% on v2.0, and 97.3% and 91.0% for neutral and non-neutral probes, respectively.

2.3.4 Comparison of the three type of approaches

- Holistic algorithms:
  Since holistic methods use global information of the face, they are usually robust for recognition over expression variations. Also, the impact of missing or noisy parts on the recognition performance is not as significant as regional or landmark-based algorithms. Moreover, face segmentation is not needed for this type of face recognition approach.

  However, the major problem with this type of recognition is the need for a reliable alignment algorithm. Generally, if the images are not registered successfully, the recognition performance can substantially decrease. These registration (alignment) algorithms also

increase the computational complexity. Another way to solve the alignment problem is to represent face in a rotational invariant space. Some examples of these rotational invariant representations are EGI [45], spin images [47] and spherical harmonics [49, 48]. However, transforming the face domain into these new representations is also computationally expensive.

In addition, the other issue with holistic methods is their storage size. Unlike other types of algorithms, which will be explained later, since the whole face is processed, the feature space size is usually large. On the other hand, this contrasts with, for example, regional algorithms where only specific parts of faces need to be stored, which is much less memory demanding.

- Regional algorithms:
  Compared to the holistic algorithms, regional approaches require smaller storage size. Moreover, their computations are performed on a subset of face surface instead of the whole. Partitioning the face is advantageous as it makes it possible to fuse the recognition scores from different parts of the face. This is very helpful, in particular for expression variations. In other words, some parts of the face are much less affected by expressions, for example the nose region [67]. Regional algorithms help to emphasize the features from these parts in order to improve the recognition performance over expression variations.

  However, regional algorithms have some disadvantages as well. In most of these algorithms, the face surface first needs to be segmented into different regions of interest. However, 3D face segmentation is a difficult problem and can be complicated when face regions are occluded by glasses, scarves, beards or moustaches. In addition, during the imaging process, some parts of the face can be self-occluded and consequently contain missing data. These situations can make it very difficult for the segmentation algorithm to successfully partition the face surface. Curvature calculation, which is extensively used in regional algorithms for face segmentation, is very sensitive to noise and imaging method (an issue that also exists for landmark-based algorithms).

- Landmark and contour-based algorithms:
  The volume of research related to this type of face recognition approach is much less than for the previously explained holistic and regional algorithms. These methods are highly sensitive to the localisations of the landmarks, which are usually found by thresholding the curvature calculation. These thresholds are usually found empirically from the gallery images and depend on the image acquisition system. Therefore, if the imaging approach changes (for example, from laser sensors to photometric stereo or the scanning resolution changes), the thresholds can be changed substantially. As a result, the

landmarks are located inaccurately, significantly degrading the recognition performance.

Moreover, detecting those landmarks which are unique for every subject is a very difficult task. In fact, a unique constellation of landmarks on the face surface to be used for recognition has not been proven yet. As an alternative to landmarks, a contour approach can be used. Although the recognition performance using the isometric contour is highly acceptable, it is very computational costly. However, the main advantage of landmark-based algorithm is their low dimensional feature vectors. This will also decrease the storage size.

2.3.5 Post-processing and Feature space creation

The features obtained from 3D faces usually have high dimensionality. This will increase the computational complexity and storage costs for large datasets. Also, some of the feature vectors might be redundant and highly correlated with each other. In order to reduce the dimension of the feature space in a way to preserve or/and increase the classes detectability, feature selection, mapping, and dimensionality reduction methods have been extensively used as post-processing algorithms on the feature space.

There are numerous feature selection algorithms used in pattern recognition problems. In this section two important categories are explained: 1) deterministic sequential methods; 2) Stochastic search-based methods. Sequential methods look for various sets of features, in a predefined order, in order to maximise an energy function at every iteration. Backward Sequential Feature Selection (BSFS) [101] initially utilises the whole feature space. Then at every iteration, the features set that when removed, reduces the recognition performance the least is discarded. This process continues until only one feature set is left. On the other hand, Forward Sequential Feature Selection (FSFS) [101] iteratively concatenates the feature sets, which produce the highest recognition performance. When the recognition performance is iteratively plotted for BSFS and FSFS, the result is usually a convex curve, whose maximum shows the optimal feature sets indexes and number.

Search-based methods, on the other hand, are non-deterministic. They search for the best combination of feature sets using a stochastic optimiser. To be more specific, an objective function is defined and maximised by selecting various combinations of feature sets. The objective function usually includes various local maxima and therefore, gradient descent-based approaches fail to find the global maximum. To overcome this problem, global optimisation approaches such as GAs or SA are used [102]. Compared to the sequential algorithms, search-based methods are much more computational complex, but are able to evaluate significantly higher numbers of feature sets combinations.

Another completely different procedure to reduce the feature space's dimensionality and

feature vectors' correlation is to utilise well-known dimensionality reduction methods. These algorithms have been extensively used in the fields of 2D and 3D face recognition. They not only reduce the features' dimension length, but also sometimes significantly improve the recognition performance. The basis of these algorithms is to find a matrix $\mathbf{W}$, with orthonormal columns, which can be multiplied by the input feature vector $\mathbf{x}$ in order to reduce its dimension from n_I to n_J, where $I >> J$.

In the PCA, which is arguably the most well-known method for dimensionality reduction, $\mathbf{W}$ is found by localising the axis, which projects the data to the direction of maximum variance. PCA is an unsupervised dimensionality reduction algorithm. Alternatively, if the samples are labeled, they can be used in the Linear Discriminant Analysis (LDA) algorithm. LDA projects the data to a lower dimensional space, in such a way that the within-class similarity and between-class dissimilarity are maximised and minimised, respectively. Both PCA and LDA project the feature space into the lower dimensional space in a linear way. Eigenfaces, are one of the most popular methods of this kind [103], is completely based on PCA. After alignment, PCA is used to find the dominant faces. The test image is then mapped on the eigen space and the closest face is recognised as the matched one.

Eigenfaces were initially proposed for 2D faces [103], but have also been used in 3D face recognition [21, 14, 104, 105]. In a slightly similar approach, PCA can be used to recognise the combination of depth and colour information [106]. This approach used different fusion methods [107] and found out that score level fusion outperforms the others. In [108], the expressions are learned using a PCA eignevectors. They are then used to re-generate the expressions on the faces. The algorithms is applied on FRGC and a 96.5% rank-one rate is achieved. Also, for the verification scenarios, 0.1% FARs of 94.05%, 98.35% and 97.8% are obtained for ROC III, neutral probe, and non-neutral probe respectively. As an alternative to using PCA, which is the basis of Eigenfaces algorithm, LDA is utilised, which has lead to Fisherfaces [109].

Despite the success of PCA in many cases, the features' distribution along the maximal variance direction can not be represented using a linear transformation. In these situations, PCA will not be capable in finding the axis of largest variance. Also, the labeled samples features might not be linearly separable. This can result in a situation, in which the linear projection using LDA can not find the most optimal projection, and, even worse, may merge some of the classes. In order to solve this issue, kernels are applied to map the feature space to a higher dimension, in which the classes can be more efficiently linearly projected. This creates the Kernel PCA (KPCA) and Kernel LDA (KLDA) or Kernel Fisher's Analysis (KFA) algorithms. Linear, polynomial, quadratic, and radial basis functions are highly popular kernels used in the literature.

Computing the matrix $\mathbf{W}$ for LDA and PCA can be computationally costly for large datasets with high feature space dimensionality. The computational complexity would be ex-

pected to be even higher for KLDA and KPCA. Also, if a new subject (or sample) is added to the dataset, the projection matrix $\mathbf{W}$ requires recomputing. Considering these issues, another approach to find $\mathbf{W}$ is to use random projection ([110]). The basic idea is that $\mathbf{W}$ is chosen independently from the gallery samples. To be more specific, as long as the columns of $\mathbf{W}$ are orthogonal and normalised (orthonormality), it can be used to project the given data to a lower dimensional space.

PCA assumes the data has a Guassian distribution and can be represented by its first and second order statistics (mean and variance) [111]. When the data is non-Gaussian, higher order statistics should be used to more properly project the data into a lower dimensional space. This is performed by ICA, which separates the input image into a linear combination of independent components [112]. Therefore, PCA is a special case of ICA, where the data is represented using the second order statistics. There are several approaches to implement the ICA algorithm. The InfoMax method for instance, finds $\mathbf{W}$ by maximising the entropy of $\mathbf{W} \times \mathbf{x}$ [111]. Other methods are JADE ([111]) and FastICA [113].

ICA is another approach used for holistic $3D$ face recognition [111]. In [114], PCA and ICA are used for 3D face recognition. The FSU 3D dataset is used for evaluation, including 222 images of 37 subjects. In [115], it has been shown that using multi-scale representation such as Gabor filter banks, Discrete Wavelet Transform (DWT) or Discrete Cosine Transform (DCT) before ICA and PCA, can significantly increase the recognition performance.

Another important operation on the feature vectors is their normalisation (also known as scaling). Each vector per sample in the feature space should be scaled in such a way that the within-class similarity is not deteriorated. The main purpose of this step is to further reduce the feature extraction's sensitivity to imaging parameters and prepare the feature space to be the input to the classifiers. For instance, the depth map values might differ based on the subject's distance from the imaging device. Therefore, if the features from the 3D facial surface are directly extracted from the depth map, they must be normalised in such a way as to maintain the similarity with the features from the same subject in other sessions.

Features normalisation in pattern recognition problems can be performed for all the samples per class. However, since the number of samples per subjects in the gallery are often not very high in biometrics datasets (usually there is only one sample per subject), the samples are usually "self-normalised". This means that the feature vectors are scaled using the information obtained from themselves. The normalisation can also occur after the matching algorithm is computed and matching scores are found. Each matching or classification criterion produces a matching score for its input feature space. Before these matching scores are merged to perform the decision making they should also be normalised. The following normalisation algorithms has widely been used in 3D face recognition problems [116, 117]:

- Unit length normalisation

$$\mathbf{x_n} = \frac{\mathbf{x}}{||\mathbf{x}||} \tag{2.1}$$

- Min-max normalisation

$$\mathbf{x_n} = \frac{\mathbf{x} - x_{min}\mathbf{o}}{x_{max} - x_{min}} \tag{2.2}$$

- Standard score normalisation (z-score)

$$\mathbf{x_n} = \frac{\mathbf{x} - \bar{\mathbf{x}}\mathbf{o}}{\sigma_x} \tag{2.3}$$

- Median and median absolute deviation normalisation

$$\mathbf{x_n} = \frac{\mathbf{x} - \text{Med}(\mathbf{x})\mathbf{o}}{\text{Med}(|\mathbf{x} - \text{Med}(\mathbf{x})\mathbf{o}|)} \tag{2.4}$$

In the above equations, $\mathbf{x}$ is the input row feature vector, $\mathbf{o}$ is a row vector with the same length as $\mathbf{x}$, whose all elements are one, and $||\mathbf{x}||$, x_{min}, x_{max}, σ_x and $\bar{\mathbf{x}}$ are its length, minimum, maximum, standard deviation, mean, and median, respectively. $\text{Med}(.)$ is the median operator and $\mathbf{x_n}$ is the normalised vector. Each of these methods has its own advantages and disadvantages and their selection is based on the feature space distribution. For instance, in the presence of outliers in the vectors, the median normaliser is more robust than the others, since it is the least affected one.

2.3.6 Matching and classification

Matching algorithms are applied on the feature vectors to produce the scores for each probe sample. There are numerous matching algorithms in the literature to compare the features in the gallery ($\mathbf{x_g}$) with the probe feature vector ($\mathbf{x_p}$). These methods can be classified as follows ($d_{p,g}$ is the matching score between the p^{th} probe and g^{th} gallery sample, $\circ$ is the component-wise Hadamard product operator and $\Sigma(.)$ computes the sum of the elements of a given vector):

- Euclidean distance

$$d_{p,g} = \sqrt{\sum(\mathbf{x_g} - \mathbf{x_p}) \circ (\mathbf{x_g} - \mathbf{x_p})} \tag{2.5}$$

- City-block distance

$$d_{p,g} = \sum |\mathbf{x_g} - \mathbf{x_p}| \tag{2.6}$$

- Cosine

$$d_{p,g}(\text{in Radians}) = \cos^{-1} \frac{|\mathbf{x_g}.\mathbf{x_p}|}{|\mathbf{x_g}||\mathbf{x_p}|} \tag{2.7}$$

- Correlation (σ_{pg}, σ_p and σ_g are the cross-covariance between $\mathbf{x_p}$ and $\mathbf{x_g}$, the variance of $\mathbf{x_p}$ and $\mathbf{x_g}$, respectively.)

$$d_{p,g} = \frac{\sigma_{pg}}{\sigma_p \sigma_g} \tag{2.8}$$

- Mahalanobis distance (Σ is the covariance matrix computed using the gallery samples $\mathbf{x_g}$.)

$$d_{p,g} = \mathbf{x_g} \Sigma^{-1} \mathbf{x_p}^\intercal \tag{2.9}$$

- Mahalanobis cosine distance

$$d_{p,g} = \left(\frac{\mathbf{x_g}}{\sqrt{|\mathbf{x_g} \Sigma^{-1} \mathbf{x_g}^\intercal|}} \right) \Sigma^{-1} \left(\frac{\mathbf{x_p}}{\sqrt{|\mathbf{x_p} \Sigma^{-1} \mathbf{x_p}^\intercal|}} \right)^\intercal . \tag{2.10}$$

The ICP algorithm can also be directly used to compute matching scores. For instance, an ICP-based matching approach is proposed in [53], where the registration error is considered for matching to range images. More specifically, three feature sets are combined to calculate the matching: ICP error obtained from the registration, colour and texture vectors, and the differences in SI calculated on both range images. The result has been tested on a dataset including 10 subjects with 63 samples. In [32], face recognition is performed by integrating 2D and 3D data. A method named spherical face representation (SFR) is introduced that quantises the 3D data by mapping the face into concentric spheres. A pattern rejection algorithm and the scale-invariant feature tranform (SIFT) are used to narrow the search domain in the gallery images. Finally, ICP is utilised for matching.

Another matching criterion used for both 3D and 2D face recogntiion problems is the Hausdorff distance. In [55], after using ICP algorithm for aligning the probe and gallery images, the Hausdorff distance is calculated for recognition purposes [118]. After performing this comparison for the whole dataset, the image pair with the lowest error is considered as the matched images. As this algorithm is highly sensitive to pose variations, a successful alignment is vital for the recognition performance. Therefore, it is stated in the paper that the Hausdorff distance calculation and alignment can be performed iteratively and the lowest value is considered as the measurement error for the probe and candidate image in the gallery. The Hausdorff distance has also been used in a slightly similar approach for recognising 3D face images [119]. Also, after fine and coarse registration, Pan *et al.* use the Hausdorff distance to compared the gallery and probe images [120]. The algorithm has been compared with eigenface approach.

As the matching algorithms usually do not require a training step, they are more beneficial for face recognition applications than the classification methods. However, classifiers have also been extensively used in this field. HMM, SVM, boosting, neural networks and, especially during the last five years, sparse representation classifiers are very popular pattern classification techniques for face recognition applications.

Achermann *et al.* used Eigenfaces and hidden Markov models (HMM) for recognising range images of faces [104]. Tsalakanidou *et al.*, Eigenfaces are calculated on both of the colour components and range data over the XM2VTS database [121] including 295 subjects, each of them with 12 images. The experiments show a remarkable improvement after using range data with colour images. Embedded HMM are also used by Tsalakanidou *et al.* for an integrated 2D and 3D face recognition algorithm [122]. The depth is applied directly to train the classifier. The algorithm is evaluated on a dataset with 3000 images of 50 subjects [123].

A literature review on the applications of neural networks in biometrics is provided in [6]. Probabilistic Neural Network (PNN) [124], Radial Basis Functions (RBF) neural networks [125], and convolutional neural networks [126] have been successfully used to recognise 2D faces with minor pose variations. Neural networks used to have the problem of high computational cost for training and the learning algorithms were also easily trapped in local minima, resulting in a low recognition performance. This has been the reason why they were replaced by the SVM classifiers (section 2.4 explains how the neural networks inefficiency issues have recently been solved by introducing deep learning methods).

A landmark-based approach is used in [127]. Eight landmarks are located around the nose region and the relative distances between them are calculated. Then, dynamic programming and SVM are used for their classification. The algorithm is tested on 100 face images from the BERC dataset. SVMs are intrinsically binary classifiers and in order to extend them for a multi-classifier, different scenarios have been introduced. For instance, the one-vs.-all scenario considers each class against the rest of the classes. The procedure is repeated for each sample. Then the classification boundaries are computed and the closest class to the probe sample is used to find the output class label. Another very convenient feature of SVMs is their capability of using kernels to map the feature space to a linearly separable higher dimension.

Combining weak classifiers to create a strong one is used by boosting algorithms, of which the most popular one is AdaBoost. Tree classifiers have been widely used for boosting methods. The basic idea behind boosting algorithms is that instead of training a strong classifier, a set of weaker classifier (with lower classification performance) are merged using an iterative method. The weak classifiers are weighted, based on their classification performance and pooled. Some examples of papers using boosting algorithm for face recognition are [128, 129, 78].

If sufficient training samples from a class are available, the test sample can be represented as a linear weighted summation of features associated with the same class in the gallery. This is the basis of sparse signal representation algorithms for which one of their first uses was introduced by the seminal paper Wright *et al.* [130]. If the test sample ($\mathbf{x_p}$) corresponds to one of the classes in the gallery, it can be represented as $\mathbf{x_p} = \mathbf{A}\mathbf{x_0}$, where $\mathbf{A}$ is a matrix whose columns are the feature vectors per gallery samples. $\mathbf{x_0}$ is a "sparse" vector, whose elements are either zero (or close to zero) except for those elements corresponding to those gallery samples that have the same label as the test sample ($\mathbf{x_p}$).

Both $\mathbf{x_p}$ and $\mathbf{A}$ are known variables, but $\mathbf{x_0}$ is unknown. In order to compute $\mathbf{x_0}$, Wright *et al.* used a method based on minimisimg $\mathbf{x_0}$'s L1-norm. By evaluating the highest values of $\mathbf{x_0}$, this method can then be used for both the verification and identification scenarios. Sparse classification algorithms rely on sufficient numbers of training samples per class, otherwise their classification performance will deteriorate [130]. In order to make the sparse classification more robust against lack of training samples, Deng *et al.* introduced a method to learn occlusion from the given dataset and artificially create it over each sample [131]. Using this method, the sparse classifier can be trained by higher number of training samples, even though some of them are artificially created. Li *et al.* also proposed a solution for the training samples number issue by assigning weights for various rectangular patches on the facial region to more properly train the sparse classifier [85]. The weights are learned by the training samples using a multi-scale normal coordinate feature space.

2.3.7 Decision making using fusion methods

Decision making is usually the final step of a biometrics algorithm. Sometimes, it is necessary to fuse measurements, features and different classifiers outputs in order to get the best result from a given feature space. Various feature selection methods should be employed in order to get the best recognition performance. Depending on when the fusion is performed, the decision making algorithms can be categorised as follows [4]:

1. sensor level: Such as using a multi-modal approach: 2D texture and colour information with 3D model;

2. feature level: After setting up the feature space, specific features are selected based on their separability. Also, the algorithms stated above, for feature space manipulation can be utilised here;

3. match score level: Feature integration is peformed according to the classification performance;

4. rank level: After finding the ranking for a set of features, these ranks are combined;

5. decision level: Used to integrate the results of classification from different decision maps.

The focus in this section is the fusion methods for the final decision making algorithm. One widely used approach for rank-level fusion is the Borda count fusion method. The output ranks for each classification method are stacked for each class label. The final class for the test sample is chosen by integrating the votes for each class. The integration can be performed by computing the sum, product or applying a non-linear function to the votes, to penalise the higher voting levels. The fusion algorithms cover a very broad range of research in the field of pattern recognition and are beyond the scope of this work. A number of excellent review papers for combining different classification outputs and rank-level fusion methods are available in [107, 132, 133, 134].

2.4 A brief notation on "Deep Learning"

The face recognition procedure explained in the previous subsections (preprocessing, feature extraction, feature space's post-processing and matching) is a classic hierarchy used to solve the pattern recognition problems. Although this procedure has shown its robustness for many applications and on various data distributions, it has drawbacks. The first one is its generalisation. As the nature of problem changes (even if the type of imaging modality of 3D face datasets changes), the algorithms decline in performance. In other words, most of the approaches are highly problem-specific and limited to the given dataset. Therefore, the algorithms are sensitive to stochastic variations, such as noise, pose variations, occlusions and deformations caused by different expressions. The methods can not be applied to very similar problems. For instance, a face detector algorithm would need to be completely changed and re-trained to be used for an animal's face detection, and even then it might not work successfully.

These algorithms are also known as "shallow" approaches [135], as they are not capable of extracting deeper abstracted concepts from the given problem and usually perform one level of feature extraction followed by a classifier. They require the machine learning and image processing expertise to look for the specific features and designing mathematical algorithms to extract the features.

Alternatively, in recent years, the increasingly high computational capacity of computers, achieved by incorporating GPUs' parallel or cloud computing, multi-layer neural networks have again become popular. To be more specific, neural networks used to have a limited pattern classification performance and their generalisation capability was not high enough. This was mostly because of the fact that most of the lengthy training algorithms usually trapped in local minima as the network size increases [136]. As a consequence, the network's weights and

biases for many neurons become zero, the classes' discrimination planes are not computed correctly, the feature selection per layer fails, and the classification rates would become too low.

This problem is now significantly ameliorated using the computational speed of current machines and resulting in the high success of the deep learning algorithms. The "deep" phrase is used against the previous "shallow" methods and arises from the fact that the classification problem is not directly applied to the given feature space. A typical deep architecture is shown in Fig. 2-7. Various levels of abstraction are computed on the images through different layers of neurons to perform feature selection ([137, 138]) and, finally, classification. The deep learning paradigm closely matches the way human brain performs automatic feature extraction and updates itself with various new and unseen samples.

The main disadvantage is that the deep architectures' training is not as quick as shallow approaches. Also, handling the over-/under- fitting issues by choosing appropriate number of neurons, layers and training epochs is still an ongoing research problem [139]. Choosing insufficient number of neurons for the hidden layers might causes the network to regenerate the same training samples (over-fitting). On the other hand, too many neurons compared to the samples numbers can leave some of the weights as zero, increase the computation and stopping the optimisation algorithm at local minima (under-fitting) [5].

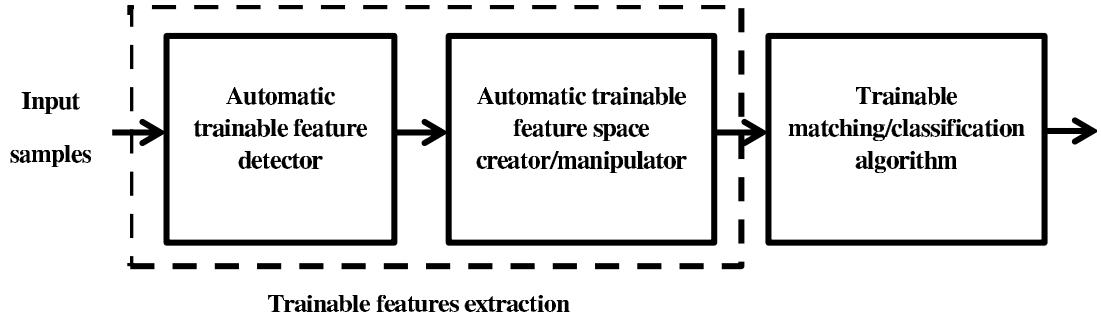


Figure 2-7: The paradigm for deep learning algorithms. Unlike shallow architectures, the feature detection, extraction, selection and classification steps are adaptively performed by trainable units.

2.5 Using the nasal region

In the previous decades, when only 2D face images were available for recognition and 3D imaging was not readily available, based on 2D data, it was believed that nose would not have enough uniqueness to be used as a biometric [18]. In particular, without having the depth information of the face surface, the frontal 2D view of the face does not provide all of the features of the nose region. However, since capturing the depth information has become much

easier due to the 3D image acquisition technology over the past decade, the potential of the nose as a biometric can be investigated more robustly and accurately.

The most fundamental reason of choosing the nose region for recognising people is its small variations in different expressions. Compared to other parts of the face, like the cheeks, eyes, forehead and mouth, it is much less changed in non-neutral expressions. Moreover, since the nose tip is usually the closest point to the camera and its convexity is more salient than other parts of the face, the nose can be easily segmented on the face. Furthermore, hiding the nose in real biometric sessions is nearly impossible, without attracting suspicion.

There are a few papers which have focused specifically on 3D nose recognition [67, 95, 140, 40]. Although, these approaches could be categorised as regional algorithms, this section will explain them in more detail. In all of these papers, the 3D information of the nose region is utilised. Chang *et al.*, first use 2D colour information for face detection, before aligning the face using ICP, and employing curvature calculation is utlised to segment the face [67]. The algorithm detects eye pits, nose tip, and bridge of the nose using curvature thresholding. Three regions are extracted from the nose region and they are matched in the gallery and probe images to perform the recognition. For the gallery image, the extracted nose region is chosen to be slightly bigger than the nose. On the other hand, smaller nose regions are obtained from the probe image. The FRGC dataset is used which includes 4000 images of 449 subjects. The EER are 12% and 23% for neutral and non-neutral probes. The rank-one recognition rates are 97% and 86.1% for the neutral and non-neutral samples.

Drira *et al.* present another approach for 3D face recognition using the nose region [95]. Their method utilises the geodesic contours, used in [94], only on the nose region. After denoising and nose segmentation (by fitting a sphere with radius 100 mm on the nose tip), the concentric contours are found on the nose surface using Dijkstra algorithm [141]. In the test phase, these contours are compared for the gallery and probe images and the closest ones are assumed to give the matched. The algorithm was evaluated on the FRGC dataset and compared with ICP, where it was found to have a better performance both for the whole face (holistic) (about 88% rank one recognition) and the nose region only (about 78% rank one recognition).

Dibeklioglu *et al.* first perform different curvature calculations on the face surface and then segment the nose region [140]. Then, the ICP algorithm is used for recognition. The Bosphorus dataset [13] is used for recognition including 4652 images of 105 subjects. The interesting feature of this 3D face dataset is its big variations in pose. The rank one recognition results presented in [140] for frontal facial images were about 94% and for pose variation is 79%.

Moorhouse *et al.* introduced another approach for 3D nose recognition [40]. Colour information in YCbCr domain is clustered by GMM and curvature calculation is used for landmark detection and nose segmentation. Unlike the previous three 3D nose recognition papers, in-

stead of using laser scanners, photometric stereo imaging is utlised to make the dataset. Four different features are evaluated on the nose surface: geometric ratios, the Fourier descriptors (FD) of the ridge, combination of these features and eigennoses. The algorithm is performed on a small dataset including 23 subjects and the highest rank one recognition obtained is around 47% by the eigennose algorithm.

Although the approach described in [89] is not necessarily on the nasal region, some experiments have been provided on the nose as well. The shape difference boosting algorithm is applied on different cropped region on the face. To perform the cropping, spheres centralised at the nose tip, are intersected with the face. For the cropped region the shape difference boosting methodology is used. For a sphere of 44 *mm* radius (which contains the nose, and some part of cheeks and mouth), a 95% rank-one recognition rate is obtained. However, for a 24 *mm* radius sphere (which mainly includes the nasal region), only a 77.5% rank-one recognition is achieved.

A combination of the nasal region, forehead and eyes are used for a 2D/3D face recognition by Mian *et al.* [32]. A modified ICP algorithm is used for matching, in conjuction with a pattern rejector based on SFR and SIFT. As a result high recognition ranks are obtained on the FRGC dataset, in particular for neutral probes. For the FRGC v2.0 datasets, 95.91% and 92.2% rank-one recognition rates are obtained for the combination of nose/forehead/eyes and nasal region, respectively. For the neutral probes, 99.2% and 94.9% recognition rates are obtained for the multi-modal and nasal regions, respectively. However, as the non-neutral samples are applied, the rank-one rates reduce to 95.37% for the multi-modal and 80.0% for the nasal approaches.

2.6 3D face datasets

In this subsection, a brief explanation about the face datasets which will be used in the experiments are provided. The first one is the FRGC, which is known as the largest 3D face dataset, with the highest number of subjects (557). The dataset was obtained using Minolta laser senors in three different seasons, Spring 2003, Fall 2003 and Spring 2004 [14]. The samples in the Spring 2003 folders are known as the v1.0, while the collection in the other two folders constitute v2.0. FRGC v1.0 and v2.0 have 267 subjects (838 samples) and 466 subjects (4007 samples), respectively. The dataset includes nearly well-aligned faces, which are much closer to the practical biometrics sessions.

In order to evaluate the face recognition algorithms on this dataset, three sets of experiments are usually defined. For the first one FRGC v2.0 is divided into 466 samples for gallery and 3541 samples for the test. This arrangement has been extensively used in the literature. The second experiment is known is FRGC's ROC III [14] on Exp III. This is a verification scenario, which uses the between-seasons 3D range data. For this experiment, usually EER or 0.1% FAR

is reported.

The third evaluation on FRGC is termed expression vs. expression. The dataset consists of samples with neutral and non-neutral expressions, while the expressions are limited to neutral, surprised and happiness. Using different sets of samples with different expression types for the probe set results in two evaluations: neutral gallery vs. neutral probe and neutral gallery vs. non-neutral probe. The purpose of this experiment is to check the algorithm's robustness against unseen expressions in the gallery.

The second dataset widely used in this research is the Bosporus 3D face dataset. It consists of 105 subjects, captured using structured-light based 3D system [13]. Compared to the FRGC, this dataset is less noisy and includes higher number of facial expressions the FRGC The Bosphorus database includes neutral, anger, disgust, fear, happiness, sadness and surprise. For this dataset two methods are usually used for evaluation. The first uses one neutral sample per subject as the gallery (105 samples) and the rest for the test phase (2797). The second approach is based on using different expressions for the gallery and other expressions as the probes.

The UMB dataset is also used in previous researches, especially for intensive expression variation and occlusion robustness evaluation [15]. This dataset consists of 143 subjects (1473 image captures), including 98 males and 45 females. It contains images with significant occlusion, pose and expression variations. The occlusions are mainly due to glasses, hair, scarves, hats, hands, or different objects. Compared to the FRGC v2.0 dataset, the images in UMB are more challenging for evaluating a pose correction algorithm, which is suitable for robustness evaluations. The Minolta Vivid 900 laser scanner is employed for obtaining the 3D data.

2.7 Conclusions

2.7.1 Summary and future work direction

Starting with a brief introduction on biometric algorithms, their important features and evaluation methods, the most recent and influential papers in the field of face recognition have been discussed. The methodology used for different steps of the algorithms was used to categorise the previously reported works. To do this, a typical 3D face recognition algorithm was explained as a hierarchy of preprocessing, feature detection, feature extraction, post-processing, feature space creation and finally, matching/classification and decision making.

Face recognition is an ongoing field of research and every year numerous algorithms are proposed to make the techniques more robust against challenging issues such as expression variations, occlusions (both self- and partial occlusions), low training samples per gallery and large datasets. Finding better models to represent different expressions or localising robust features invariant to expressions is still an open research topic. Automatically detecting the occluded parts of the face using machine learning algorithms would also be a very valuable

capability for a biometric modality. Designing an algorithm that is insensitive to the type of imaging modality or to very low resolution images (such as the ones obtained from CCTV cameras) would also enable biometric users to apply the systems to more diverse datasets.

Although most of the previous "shallow" architectures for face recognition are mainly focused on fixed feature extraction and classification steps, recent "deep" architectures, are concentrated on "trainable" feature extraction and classification design. With the rapid advances in the computational speed during the recent years, the use of random sampling, compressive sensing and in particular, deep learning methods have been gradually making the feature extraction step less important. The most challenging side, however, is on the classification and feature learning stages.

Key factors are how to more properly quantify sparsity for the sparse classifiers, design deep network architectures with faster more robust learning algorithms (both the optimisation procedure and objective functions), that are less vulnerable to local minima, and the formulation of the number of neurons and layers for a deep neural network. Also, one of the most important aspects of the deep architectures is to understand the mathematics and type of features, which are automatically learned and extracted from images at every layer. This would enable the localisation of the most discriminantive features on the face (and also, for other biometrics) to enhance the imaging modality on those parts or even storing those sections as the biometric data.

2.7.2 Ongoing challenges

The effect of the choice of denoising algorithms' parameters for the 3D face recognition performance have not been specifically addressed by previous research. In particular, if the imaging modality changes and, as a consequence, the noise distribution changes then the denoising, and hence face recognition, might fail to operate successfully. Therefore, modelling the noise can help to detect the most vulnerable facial regions to the noise and could also help the simulation of noise to perform a robustness evaluation for both denoising and face recognition approaches. This topic is addressed in chapter 3.

3D face detection and segmentation, which is usually performed after the nose tip detection, can have significant influence on the subsequent face recognition steps. It can directly influence the pose correction and feature vectors correspondence. Occlusions, pose variations and facial expressions can deteriorate the quality of the nose tip detectors and hence the face region cropping. Also, the facial alignment can be sensitive to these issues. Failure to have a consistent, accurate and robust alignment algorithm can significantly alter the feature distribution for the subjects and can result in misclassification. To address these issues, chapters 4, 5 and 6 present a new nose tip detection and robust pose correction algorithms using the nasal regions, respectively.

Landmarking algorithms are usually very sensitive to the deformations caused by the facial expressions. If the landmarking step is not robust against expression variations, the feature descriptors, which are calculated around the landmarks can be detected wrongly. The feature space also needs to be capable of handling minor variations in pose, noise and facial expressions. Another important challenge is to be able to recognise the probe images when the gallery size is very small, and in particular, using only one training sample per subjects in the gallery. Chapters 7 and 8 present new landmarking algorithms and feature descriptors to be used for robust 3D face recognition using the nasal region, when a single training sample is used. These chapters reveal the nasal region distinctive features for biometric applications.

Chapter 3

Denoising

3.1 Introduction

The use of the 3D face surface for human recognition has significantly grown in popularity during the past decade. One of the most important reasons behind this is the improvements in the 3D imaging and depth map generation technologies. Imaging devices are becoming cheaper and more user-friendly, enhancing their attractiveness for consumers in the face recognition market. However, like any other electronic device, noise is an inevitable part of the 3D imaging systems. In addition to additive white noise, 3D depth maps are also degraded by spike noise, holes and missing data. These issues directly affect the performance of 3D face recognition systems.

A typical 3D face recognition system consists of the following steps: preprocessing, alignment or registration, normalisation and matching. Denoising, which is typically performed as part of the preprocessing, can have a significant affect on the subsequent stages. For example, if the facial data includes spike noise, the pose correction or registration step might fail. Also, since the data should usually be normalised before being added to the feature space, noise can completely change the normalised vector formation. As a consequence, the feature vectors might not be correspondent and/or outliers might be added to the feature space, resulting in significant degradation of the classification performance.

Noise can have a different effect in the test phase. For example, in a landmark-based face recognition algorithm noise can degrade the landmarking algorithm during authentication such that the localisation of landmarks on the subject's face can differ from those already stored in the gallery. Noisy biometric data can also result in false inter-class similarities that in extreme cases can cause the subject might to be misclassified. To overcome these problems, some researchers have tried to improve the robustness of face recognition algorithm by to finding automatic methods to evaluate the quality of samples in both the gallery and probe datasets.

For instance, in [142] and [143], automatic methods are provided to measure the quality of the 2D images and 3D depth maps of faces, respectively. The low quality samples are rejected and not considered as a valid input.

In summary, in 3D face recognition (or any other biometric system) the major issue caused by noise is the reduction in the within-class similarity and the between-class dissimilarity. In this chapter, the effect of the choice of various denoising algorithms on the performance of baseline holistic 3D face recognition algorithms is analysed. First, the face dataset's gallery images are divided into a training and query subsets and the denoising algorithms are applied over the samples. Then, the holistic face recognition algorithms, which use the whole facial region as the feature vectors [18], are evaluated on the feature space. The classification methods selected are well-known and have been extensively used in the field of facial biometrics. The recognition ranks are computed and the best denoising technique is chosen based on the classification performance.

Using the face recognition performance, the evaluation process can also be used to tune the denoising algorithms' parameters. The conventional expectation is that if the denoising is too aggressive the loss of high frequency components will adversely affect the recognition performance. To avoid information loss in the parts of the face critical for matching, the aggressiveness of the denoising algorithm should be limited or correctly tuned. However, results show that for many denoising algorithms this limit is much higher than that traditionally used. Also in this chapter, the effects of varying the noise power on the 3D holistic face recognition algorithms are evaluated. First, instead of manually applying Gaussian noise to the face surface, as used in [144, 145], an algorithm is proposed to learn the noise distribution from the 3D faces and simulate it on any given face. The method consists of finding the eigenshape of the difference maps computed over the aligned face of each subject. Then, a probability map is defined and used to model the noise. The method provides the capability to gauge the denoising performance on the face recognition algorithms. This is one the first studies on the denoising effects on 3D face recognition performance. The proposed algorithm can be applied to any 3D facial dataset to improve the denoising robustness and parameters tuning.

The chapter is organised as follows. The face recognition procedure used to evaluate the denoising performance is described in Section 3.3.1. Then, the noise learning and simulation procedures are explained in section 3.3. Experimental results are given in Section 3.4. Finally, a summary and discussion are provided in Section 3.5.

3.2 3D face recognition pipeline

Holistic 3D face recognition systems consist of face cropping, pose correction, normalisation and matching steps, as shown in Fig 3-1. The denoising algorithms are described in detail in

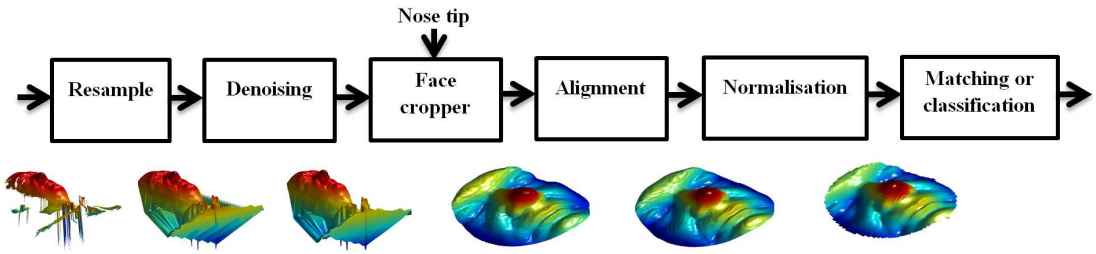


Figure 3-1: Block diagram of the face recognition system used in this work to evaluate and find optimal parameters of the denoising algorithms, in order to have higher recognition results.

Section 3.4; in this section the methods used before and after denoising are briefly explained.

When the 3D data is acquired, it is assumed that it contains both the face and upper-chest region (possibly the neck and shoulders). The noise in the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps is much less marked than that in the $\mathbf{Z}$ depth map [32, 2, 146]. Therefore, here the denoising is only applied over the depth image. To perform this task, after acquiring the 3D raw data, resampling is applied to replace the noise and missing data in the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps. The nose tip is relocalised on the resampled data and, after temporarily replacing the invalid points' depth value with the median of the valid points' depth, the denoising algorithm is applied. A sphere, centered on the nose tip, with radius 80 mm is then intersected and the facial region is cropped. The face is then aligned using the iterative PCA algorithm [32]. Finally, the resulting depth image is resized and scaled to the range of $[0, 1]$. This produces the feature space which is used in the matching step.

Using this pipeline to quantitatively evaluate the effects of denoising has many advantages. First, the denoising performance is not only quantified on the matching, but also on the alignment and normalisation. This is a legitimate assumption, since in reality the noise can add outliers and missing data to the feature space which can reduce the matching and recognition performances. For example, spike noise can have a significant influence on the pose correction step. Also, normalisation can be very vulnerable to the impulsive variations in the depth map due to the noise. Failure in both of these parts can degrade the feature vectors correspondence and recognition performance.

3.3 Noise modelling using a probability map

The noisy samples in the face datasets can be used to find a model for the noise which can then be used to simulate noise over other, less noisy samples. First, it is assumed that the i^{th} subject ($i = \{1, \ldots, N\}$) in the dataset has J_i non-occluded samples with neutral expression. These samples are resampled using the Delaunay triangulation to interpolate the missing data and remove the noise in the coordinate maps. This operation results in a set $\{\mathbf{F}_{i,1}, \mathbf{F}_{i,2}, \ldots, \mathbf{F}_{i,J_i}\}$

where each $\mathbf{F}_{i,j} = [\mathbf{X}_{i,j}, \mathbf{Y}_{i,j}, \mathbf{Z}_{i,j}]$ is a $M \times 3$ matrix whose columns correspond with the x, y and z axes points. Then, using a $2.5mm \times 2.5mm$ mask, a median filter is applied twice to remove the spike noise and smooth the surface. The faces are then aligned using singular value decomposition over the 3×3 covariance matrix $\Sigma_{i,j}$ for the j^{th} sample of the i^{th} subject ([32]),

$$\mathbf{V}_{i,j}^{-1} \Sigma_{i,j} \mathbf{V}_{i,j} = \mathbf{E}_{i,j}, \tag{3.1}$$

where $\mathbf{V}_{i,j}$ and $\mathbf{E}_{i,j}$ are the 3×3 eigenvector and eigenvalue matrices, respectively. $\mathbf{V}_{i,j}$ is used as a rotation matrix and multiplied by the noisy input point clouds after their averages, $\mathbf{m}_{i,j}$ (a 1×3 vector), are removed. The result is the aligned set of point clouds, $\mathbf{F'}_{i,j}$, given by ($\mathbf{I_o}$ is a $M \times 1$ column vector, whose elements are one)

$$\mathbf{F'}_{i,j} = (\mathbf{F}_{i,j} - \mathbf{I_o} \mathbf{m}_{i,j}) \mathbf{V}_{i,j}. \tag{3.2}$$

After this operation, all the J_i noisy samples of the i^{th} subject are aligned. As $\mathbf{F'}_{i,j}$ will be the depth map independent from the resolution information, its first and second columns, which contain the x and y axes data, are discarded. This makes $\mathbf{F'}_{i,j}$ a $M \times 1$ vector. The Principal Component Analysis (PCA) algorithm was not applied directly to the original noisy point cloud data because of its significant vulnerability to spike noise and outliers in the data. Performing the eigenvalue decomposition on the median filtered point clouds and updating the noisy faces afterwards overcomes this issue.

The aligned faces are all resized to a fixed height and width and the difference map is then accumulated using the aligned faces ($\circ$ is the Hadamard's component-wise product) ,

$$\mathbf{D}_i = \sqrt{\sum_{m,n,m \neq n} (\mathbf{F'}_{i,m} - \mathbf{F'}_{i,n}) \circ (\mathbf{F'}_{i,m} - \mathbf{F'}_{i,n})}. \tag{3.3}$$

where $\mathbf{D}_i$ is the difference map for the i^{th} subject. This procedure is repeated for all the subjects in the dataset, giving the difference matrix $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_N]$. In order to find the difference map which contains the maximal shape variation between the difference maps, the PCA algorithm is applied to $\mathbf{D}$, whose covariance matrix is the $N \times N$ matrix Σ_{D}. In other words, the N-dimensional space is projected to one dimension, in which the maximum variance of the data distribution is preserved, by

$$\begin{cases} \mathbf{V}_{\mathrm{D}}^{-1} \Sigma_{\mathrm{D}} \mathbf{V}_{\mathrm{D}} = \mathbf{E}_{\mathrm{D}} \\ \bar{\mathbf{D}} = (\mathbf{D} - \mathbf{I_o} \mathbf{m}_{\mathrm{D}}) \mathbf{U} \end{cases} \tag{3.4}$$

where $\mathbf{m}_{\mathrm{D}}$ is a $1 \times N$ vector including the average of the rows of $\mathbf{D}$ and $\mathbf{U}$ is a column vector including the eigenvector corresponding to the highest eigenvalue, which is obtained from $\mathbf{V}_{\mathrm{D}}$.

$\bar{\mathbf{D}}$ is the "eigen-"difference shape, which includes the highest variances of $\{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_N\}$. $\bar{\mathbf{D}}(x, y)$ shows the significance of the noise strength at point (x, y). If the area around (x, y) is flat, then the noise power will be very low over the area. Alternatively, peaks or valleys in $\bar{\mathbf{D}}$ correspond to a higher probability of having spikes or holes in that particular region of the face. Since all the images used to create $\bar{\mathbf{D}}$ have a neutral expression, are non-occluded and aligned, the only difference between them is the noise. Consequently, the values of $\bar{\mathbf{D}}$ at different points provide a good measure to identify which parts of the face are noisier after the 3D data capture.

A major, additional value of $\bar{\mathbf{D}}$ is that it can be used to create a probability map which can be used to artificially simulate noise over any set of 3D face data. This will enable the noise power to be stochastically varied thus underpinning the quantitative evaluation both the de-noising performance and the robustness of face recognition algorithms. The noise occurrence probability map, $\mathbf{P}$, for each pixel (x, y), is simply found by normalising $\bar{\mathbf{D}}$ to the range 0 to 1, using the min-max normalisation,

$$\mathbf{P} = \frac{\bar{\mathbf{D}} - \min_{x,y}(\bar{\mathbf{D}})}{\max_{x,y}(\bar{\mathbf{D}}) - \min_{x,y}(\bar{\mathbf{D}})} \tag{3.5}$$

Any face data in the dataset can now be degraded by adding a randomised simulated noise map $\delta\mathbf{F}$ to its depth image,

$$\begin{cases} \mathbf{F}^n_{i,j} = \mathbf{F}_{i,j} + \delta\mathbf{F}_{i,j} \\ \delta\mathbf{F}_{i,j}(x, y) = \begin{cases} a\bar{\mathbf{D}}(x, y), & \text{if } \mathbf{P}(x, y) \geq \mathbf{r}_{i,j}(x, y) \\ 0, & \text{if } \mathbf{P}(x, y) < \mathbf{r}_{i,j}(x, y) \end{cases} \end{cases} \tag{3.6}$$

where $\mathbf{r}_{i,j}$ is a matrix with the same size as $\mathbf{P}$, whose elements are randomly assigned, for the j^{th} sample of the i^{th} subject, using a uniform distribution in the range $[0, 1]$, $\mathbf{F}^n_{i,j}$ is the noisy face image and a is a scalar used to vary the noise power. When $\mathbf{P}(x, y)$ is high (e.g. at fragile, noisy parts of the face), its value will be more likely to be higher than the uniformly selected random number $\mathbf{r}_{i,j}(x, y)$ and the noise at that point is amplified. On the other hand, lower values of $\mathbf{P}(x, y)$ (which correspond to less noisy facial parts after 3D reconstruction), reduce the probability that $\mathbf{P}(x, y) \geq \mathbf{r}_{i,j}(x, y)$ and hence of additional noise being added to the data at that point. The whole procedure is depicted in Fig. 3-2.

3.3.1 Face recognition methods evaluation pipeline

The evaluation employs holistic 3D face recognition algorithms, which have been widely used for both 2D and 3D face recognition. In the first step, all faces are cropped and aligned. As the noise in the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps is much less marked than that in the $\mathbf{Z}$ depth map

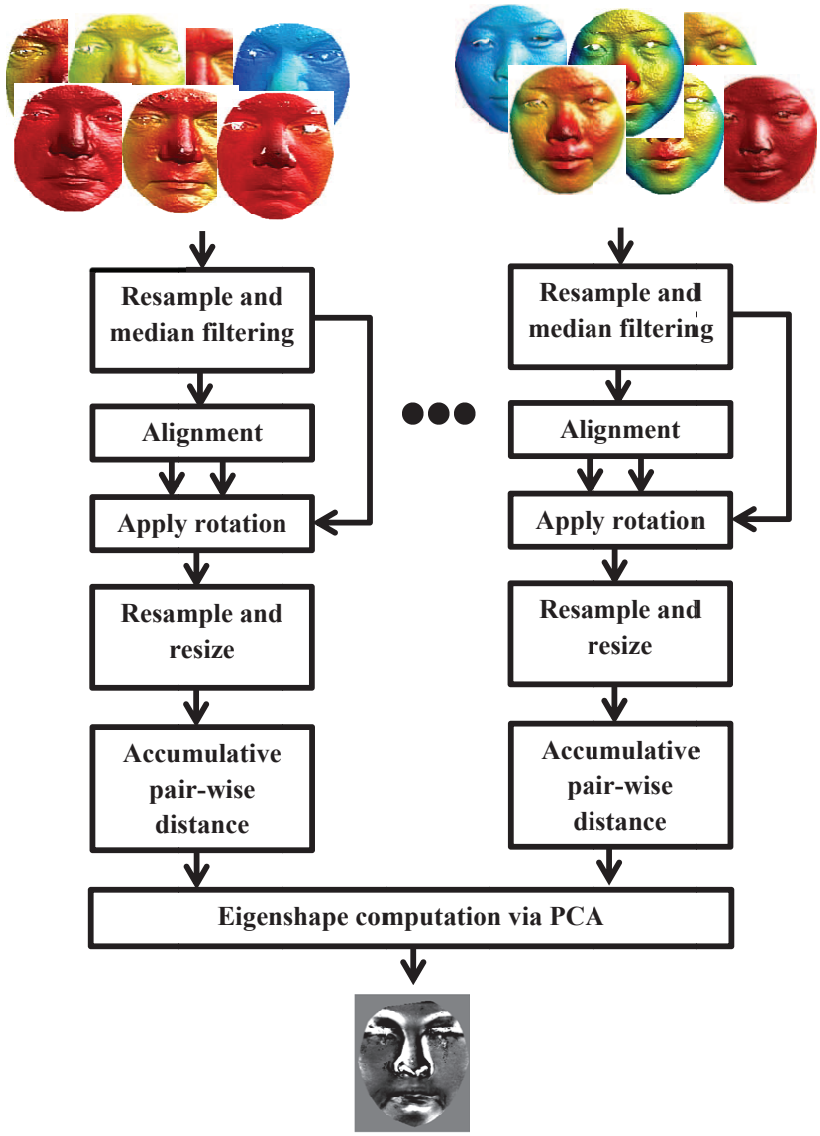


Figure 3-2: The procedure of finding $\bar{\mathbf{D}}$: After resampling and alignment, the mean vectors and pose rotation matrices are computed per sample, which are then applied over the input noisy point clouds to compute accumulative difference map $\mathbf{D}_i$. The eigen-difference shape $\bar{\mathbf{D}}$ is eventually calculated over $\mathbf{D}_i$.

[32, 2, 146], here the denoising is only applied to the depth image. To perform this task, after acquiring the 3D raw data, resampling is applied to replace the noise and missing data in the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps. The nose tip is relocalised on the resampled data and, after temporarily replacing the invalid points' depth value with the median of the valid points' depth, the denoising algorithm is applied. Then, a sphere, centered on the nose tip, with radius 80 mm is then intersected and the facial region is cropped.

The depth maps are resized to the same size as the eigen-difference shape $\bar{\mathbf{D}}$ and the noise power is tuned using a in (3.6). Simulated noise is added to the depth maps and then the denoising algorithms described in the previous section are applied. The parameters of the denoising algorithms are set to those that gave the best 3D face recognition performance in [26]. After denoising, the depth values are normalised using the min/max normalisation of (3.5) and the resulting map is again resized. For all the resizing steps, the face is pre-filtered to avoid the high frequency aliasing and cubic interpolation is used for resampling. Finally, the resulting samples are divided into gallery and probe samples and the resulting feature vectors used by the classification or matching algorithms. The stages of the recognition pipeline are illustrated in Fig. 3-3, for two example faces.

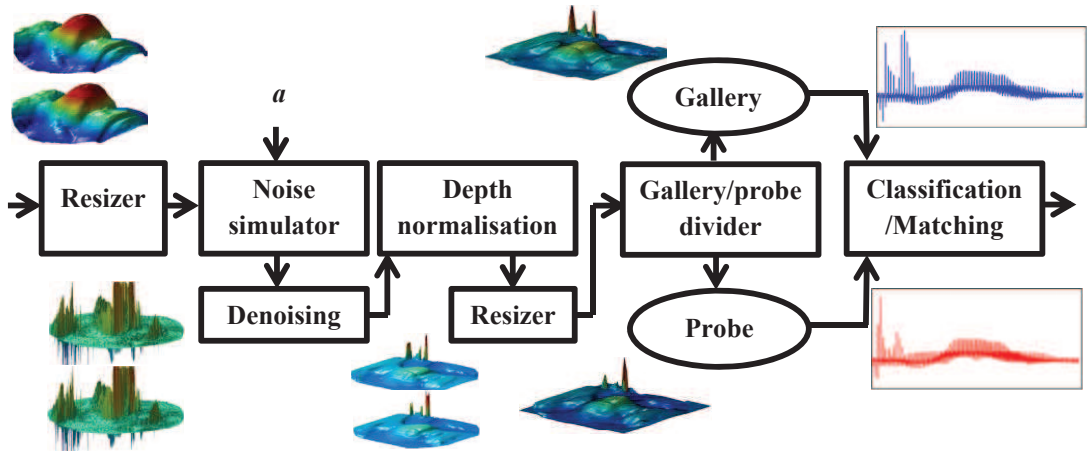


Figure 3-3: The 3D face recognition pipeline, including the noise simulation procedure ($a = 0.5$ for the simulated noisy images). After resizing, noise is simulated over the input depth maps using a in (3.6). Then the denoising algorithm is applied and the feature vectors are created after normalisation and resizing.

The face recognition pipeline of Fig. 3-3 enables the robustness of different face recognition algorithms against varying noise power to be quantified. This approach also enables the best performing denoising algorithms, in terms of recognition performance to be identified.

3.4 Experimental results

3.4.1 Dataset

The proposed algorithm for evaluating the performance of the denoising approaches and tuning their parameters is independent of the 3D face dataset used. The results below are obtained using the well-known Face Recognition Grand Challenge (FRGC) v2.0 dataset [14].

Unlike more recent face datasets, such as the Bosphorus dataset [13] and UMB-DB [15],

post-processing has not been performed on the FRGC's samples, making FRGC v2.0 more suitable for the assessment of denoising algorithms. To further appreciate the noise effects, only the images in the Spring 2003 folder are considered in the experiments, as these images are noisier than those in the other folders (as they were captured with an earlier scanner model). Those classes with at least four samples in the folder are used, giving a dataset of 119 subjects (classes) and 661 samples. Two samples per class are selected as the training (gallery) and the remaining samples as probe images.

3.4.2 The recognition algorithms

Seven popular classification algorithms are used; multi-class SVM (Multi-SVM), PCA, KFA, PNN, KNN-classification, bootstrap aggregation decision trees (TreeBagger) and LDA. For the Multi-SVM classifier a linear kernel is used. Also, the one-vs.-all scenario is utilised to transform it to a multiple-class classifier. For the subspace projection approaches [PCA (Eigenfaces [103]), LDA (Fisherfaces [109]) and KFA] the feature space is projected onto a 100-dimensional space. The polynomial kernel is employed for the KFA algorithm and the city-block distance is used for the KNN classifier ($K = 2$). Finally, 119 trees are aggregated to create the TreeBagger classifier. Matlab's Statistics and Neural Networks Toolboxes are used to implement the Multi-SVM, TreeBagger and PNN classifiers, while the PhD toolbox (Pretty helpful Development functions for face recognition [147]) is utilised to perform the subspace projection (PCA, LDA and KFA) and KNN classification.

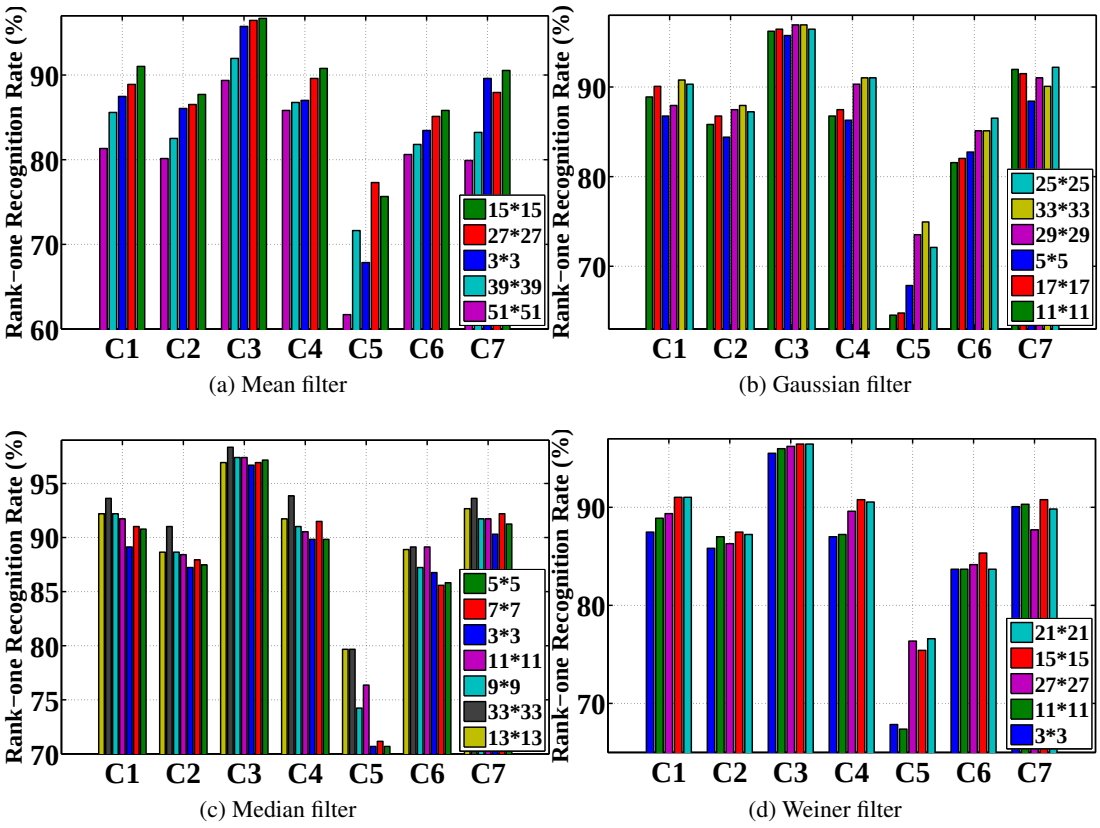
3.4.3 Denoising algorithms

Six denoising algorithms are evaluated using the dataset: Gaussian, mean and median filtering, multi-scale wavelet denoising, adaptive Wiener filtering and non-linear diffusion. The methods and their parameters are briefly explained below. For all filters the masks sizes are given in pixels as the faces are resampled using a uniform grid with a 0.5 mm/pixel horizontal and vertical resolution, as discussed above.

Gaussian, mean, median and wavelet filtering

The impulsive variations in the depth map are concentrated in its high-frequency components. Low-pass filtering is the most common way to remove these components, for example by convolving Gaussian filters ($\mathbf{G}_{\sigma}^{M}$) with the image, where σ is the standard deviation and M is the square root of the mask size. Similarly, the mean filter can also be used to remove other high-frequency noise and smooth the face surface. For both filters, the size of the mask is varied and the recognition results recorded. Median filtering has also been used extensively

for preprocessing 3D faces and is also evaluated. The results of Gaussian, mean and median filtering are shown in Fig. 3-4.



(a) Mean filter (b) Gaussian filter

(c) Median filter (d) Weiner filter

Figure 3-4: The rank-one recognition: C1: Multi-SVM, C2: PCA, C3: KFA, C4: PNN, C5: KNN, C6: TreeBagger and C7: LDA

For both the Gaussian and median filters the rank-one recognition rates increases as the mask size becomes larger. For instance, the Multi-SVM classifier's rank-one recognition rate approximately increases 1.4% when the median filtering mask size is enlarged from 13×13 to 33×33. This mask size is significantly larger than that used in previously work. Although the image is very blurred with such a large mask, the within-class similarity and between-class scatter are not only more appropriately preserved, but also improved in the feature space. A similar conclusion could be drawn from the Gaussian filtering results shown in Fig. 3-4-b. However, the mean filtering recognition performance rapidly decreases as the mask size increase. This is mainly because the background and foreground data are merged and loss of edges. Similarly, the recognition rates of the median filtered faces gradually reduces as the mask size increases.. For instance, the KFA classification's rank-one rate decreases to 96.93% and 94.33%, when 43×43 and 55×55 masks are used respectively.

Wavelet	$L=1$	2	3	4	5	6	7	8	9	10
Daubechies	96.0	96.0	95.7	96.0	96.2	96.0	96.0	96.0	96.2	96.0
Haar	95.0	95.5	95.7	95.5	96.0	96.0	95.7	95.7	95.7	95.7
Reverse Biorthogonal	95.7	95.7	95.7	95.5	95.7	95.7	95.5	96.0	96.0	95.7
Symlets	95.0	95.5	95.7	95.5	96.0	96.0	95.7	95.7	95.7	95.7
Biorthogonal	95.3	95.5	95.5	95.5	95.3	95.5	95.3	95.5	95.7	95.7
Discrete Meyer	95.7	96.0	95.7	96.2	96.45	96.0	96.45	96.0	96.2	96.0
Coiflets	95.3	95.0	94.8	95.7	95.0	95.0	94.6	94.6	95.3	95.0

Table 3.1: The rank-one recognition rates of the KFA classifier over the denoised faces using different wavelets and L levels.

Weiner filtering

Weiner filtering is a type of adaptive denoising algorithm which uses the statistical information of the input image. An $M_w \times M_w$ neighbourhood is used to estimate the noise's statistical parameters, such as variance and mean. By varying M_w, the aggressiveness of the denoising is changed and a wider image area is analysed. The rank-one recognition rates for different mask sizes are shown in Fig. 3-4-d. Although the face is again blurred with larger mask sizes, the performance of most classifiers increases up until 15×15 and the KFA classifier produces its highest recognition rate of nearly 96.22% when the mask size is 27×27.

Table 3.1 shows the recognition rates, when the wavelet filtering is applied as the denoiser. As the layers multi-resolution decomposition of the wavelet filters are increased, no significant change is detected in the rank one recognition rates. However, the Discrete Meyer wavelet results in a slightly higher recognition rate than other wavelets.

Non-linear diffusion

Non-linear diffusion is a method introduced by Perona and Malik [148] for image simplification, denoising, segmentation and feature extraction. Its concept is based on the heat transmission between adjacent materials. The diffusion's partial differential equation is iteratively solved over the image domain. If the parameters are appropriately tuned, the result will be a denoised version of the input image. The most interesting feature of the non-linear diffusion is its edge preservation, while smoothing the adjacent regions. This means that, for example, high-frequency noise can be removed from the forehead while the edges close to the eyes are maintained. Following [149], the diffusion equation used here is,

$$\frac{\partial \mathbf{Z}}{\partial t} = \nabla.(g_m(|\nabla \mathbf{Z}|^2)\nabla \mathbf{Z}), \tag{3.7}$$

in which $\mathbf{Z}$ is the face's depth image, $\nabla.()$ and ∇ are the divergence and gradient operators, respectively. $g_m(.)$ is a decreasing function [149] given by,

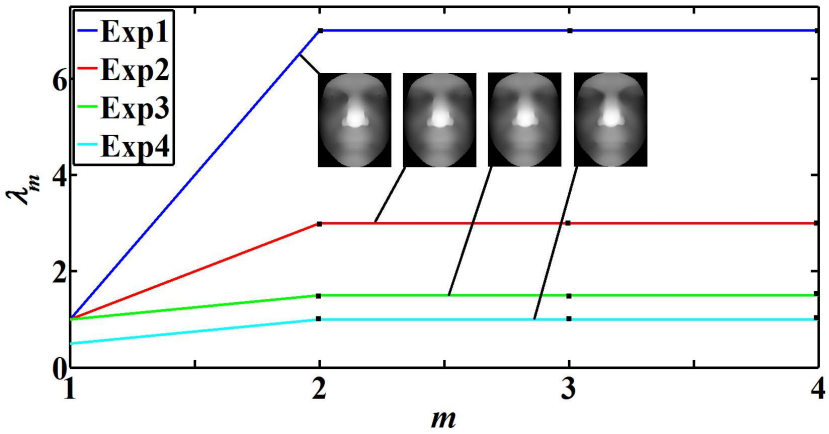


Figure 3-5: Sets of λ_m for iterations $m = 1$ to 4.

$$g_m(s) = \begin{cases} 1 - \exp(-(\frac{\lambda_m}{s})^m), & \text{if } s > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

where λ_m is the image contrast control parameter in the m^{th} iteration of the diffusion equation. The values of λ_m used in different iterations are plotted in Fig. 3-5 and the rank-one recognition rates after applying the denoising parameters are shown in Fig. 3-6. As λ_m increases, the diffusion result is more significantly blurred (see Fig. 3-5). However, the between-classes discrimination is better preserved and the rank-one recognition rate increases for all classifiers with the KFA classifier achieving the best result of 96.45%. For the implementation of the nonlinear diffusion denoiser, the "Nonlinear Diffusion Toolbox" provided by F. D'Almeida is employed [150].

3.4.4 Denoising methods comparison

Among the different classifiers used in this work, KFA with the polynomial kernel has the best performance. Its rank-one recognition rates are high for every denoising method used and it can be used to identify which denoising algorithm performs better, in terms of preserving the between/within classes scatter. The results in Fig. 3-7 show the highest recognition rank-one recognition rate for each denoising algorithm. Median filtering, with a 33×33 mask has the best performance with 98.35% rank-one recognition rate. Then the Weiner and Gaussian filters with the 19×19 and 37×37 masks, respectively, outperform the other denoising methods.

The reason for the median filter's superiority is its edge preserving performance and spike removal, while simultaneously smoothing the facial surface. The mean filter and non-linear diffusion results are 96.93% (with a 19×19 mask) and 96.45% (Exp4 in Fig. 3-6), respectively. Although the non-linear diffusion's recognition rank is slightly lower than other denoising

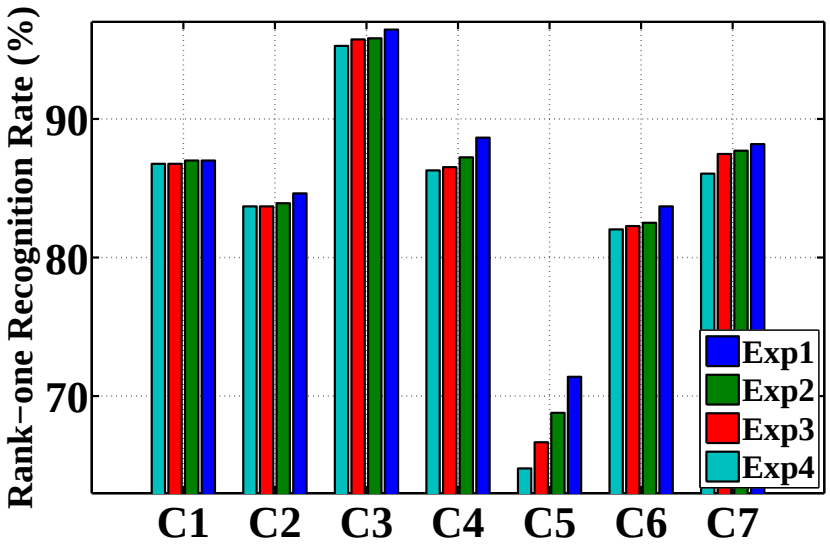


Figure 3-6: The rank-one recognition performance for the different sets of λ_m shown in Fig. 3-5.

methods, it has many parameters which can be tuned to increase its performance, such as the number of iterations or λ_m. However, since the partial differential equation needs to be solved on the image domain, it has the highest computational cost. The results of the wavelet denoising methods are significantly lower than the other methods and the best performing wavelet from Table 3.1 is worse than the other methods. All the mentioned rank-one recognition rates are clearly higher than the noisy (not-denoised) faces classification result, which is 94.80%.

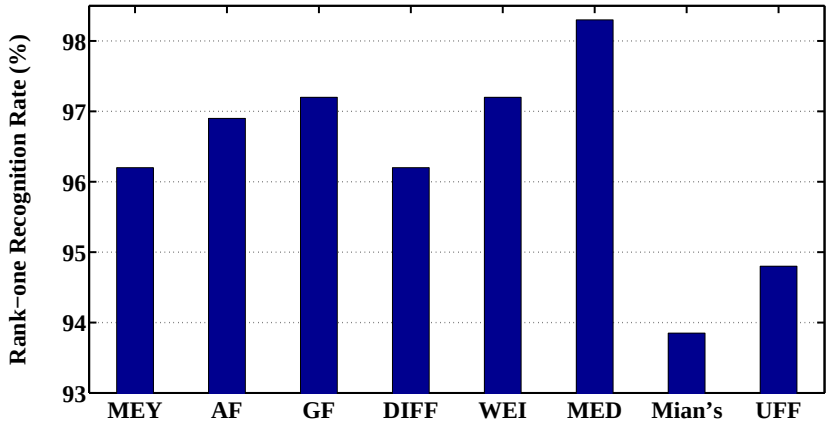


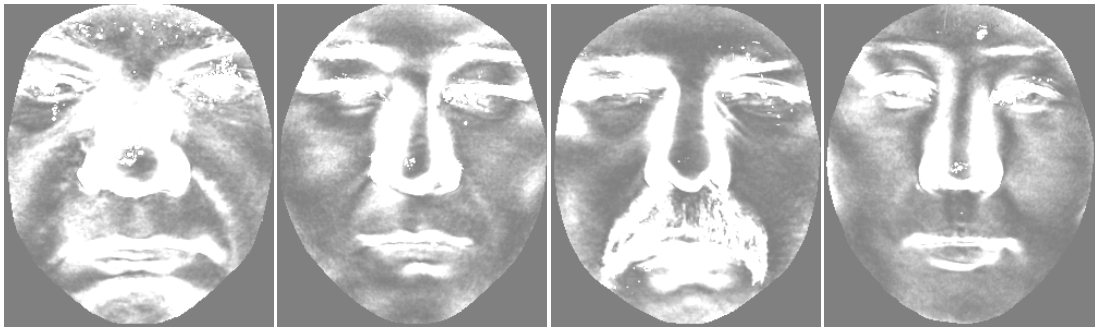
Figure 3-7: KFA classification result over the following denoising algorithms: MEY (Discrete Meyer), AF (Average or mean filtering), GF (Gaussian filtering), DIFF (non-linear diffusion), WEI (Weiner filtering, MED (median filtering), Mian *et al.* [32] and UFF (unfiltered faces).

The results are also compared with the denoising approach detailed in [32]. In their approach the mean (μ) and standard deviation (σ) of a 3×3 neighbourhood are calculated and

if the value of the central pixel is greater, it is considered to be a spike and replaced by cubic interpolation. Median filtering is also applied to smooth the surface. Using the KFA classifier, the performance of Mian *et al.*'s denoising [32] is 93.85%, nearly 4.5% below the best performing method, used in this work.

3.4.5 Noise model computation

To compute the eigen-difference shape $\bar{\mathbf{D}}$, if the number of samples per subject is too small, the resulting difference maps ($\mathbf{D}_i$) will not be accurate enough. Also, if it is too high, since there are not too many samples per subject for most of dataset's subjects, there will not be sufficient number of subjects to be able to accurately find $\bar{\mathbf{D}}$. Based on observed results, it was found that selecting those subjects from the FRGC 2003 folder with at least seven samples was a good choice. This results in a subset consisting of 28 subjects with 209 samples in total. Example shape difference maps $\mathbf{D}_i$ for four subjects are shown in Fig. 3-8-a to -d, in which regions that are more different from the neighbouring pixels have higher grey scale values. These differences are caused by the noise, since the captures used have neutral expressions and so do not have occlusions. It is interesting to see that regions located on the eyebrows, eyes, surrounding nasal region and mouth have the highest vulnerability to the noise, as the within-class similarity is at its lowest on these parts. The high frequency of the depth variations over these regions caused higher errors in the 3D reconstruction while the flatter parts the face, such as the forehead, cheeks and chin, are less noisy.

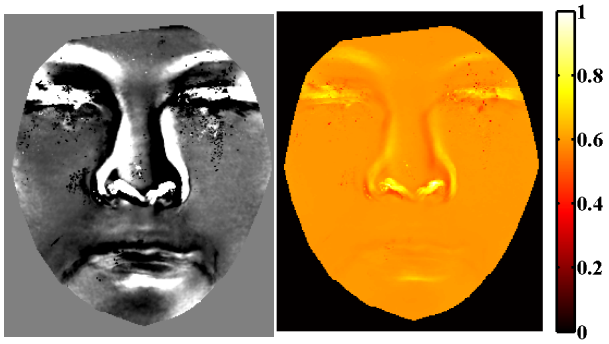


(a) (b) (c) (d)

Figure 3-8: Shape differences maps for four different subjects: ($\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4$). The brighter regions show the more vulnerable parts of the face to the noise.

The eigen-difference shape ($\bar{\mathbf{D}}$) computed over all the difference maps is plotted in Fig. 3-9-a. $\bar{\mathbf{D}}$ contains the maximal shape variations among the difference maps $\mathbf{D}_i$ and shows that the nostrils, nose tip and sides are very sensitive to the reconstruction noise. These are regions where the influence of the denoising methods will be more obvious and for the nose tip, which

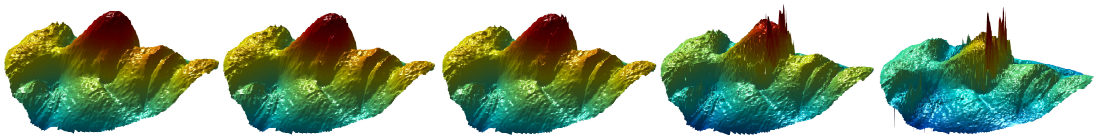
is used for face segmentation, inaccurate denoising can lead to incorrect facial region cropping.



(a) (b)

Figure 3-9: (a) The eigen-difference shape result ($\bar{\mathbf{D}}$), and (b) the probability map ($\mathbf{P}$).

Normalising $\bar{\mathbf{D}}$ using (3.5) results in the probability map ($\mathbf{P}$) shown in Fig. 3-9-b. Those regions on $\mathbf{P}$ which have higher values are more likely to be affected by noise. Using $\mathbf{P}$ in (3.6) to simulate noise on the aligned depth map for an example image produces the noisy images shown in Fig. 3-10. Figure 3-10-a shows the aligned depth map and Fig. 3-10-b to -e show the noisy images as the noise power is increased by varying a in (3.6). As the noise power is intensified, the spike, holes and high frequency noise will become more salient in the depth maps.



(a) (b) (c) (d) (e)

Figure 3-10: (a) An example of the input depth map $\mathbf{F}_{i,j}$ in (3.6) and the resulting $\mathbf{F}^n_{i,j}$ for: (b) $a = 0.0005$; (c) $a = 0.001$; (d) $a = 0.005$ and (e) $a = 0.01$.

3.4.6 Denoising methods performance

The use of noise modelling to simulate the noise over different depth maps enables a quantitatively evaluation of the success of different denoising and classification/matching algorithms to be performed. To this end, the seven widely-used denoising methods from section 3.3.1 are used to recognise faces. These are the m-SVM, PCA, KFA, PNN, TreeBagger, LDA and KNN classifier, with three different distance criteria: cosine (Cos), Euclidean (Euc) and city-block (Ctb).

Denoising methods	**Parameters**
Gaussian filtering	$M = 37$ mask and $\sigma = 6$
Median filtering	33×33 mask
Mean filtering	19×19 mask
Non-linear diffusion	Exp4
Weiner filtering	$M_w = 19$
Wavelet filtering	Five levels ($L = 5$) of the discrete Meyer wavelet decomposition

Table 3.2: The configuration for the denoising algorithms. These parameters produce the highest rank-one recognition rates, when applied to the input faces in the dataset used in this work.

The result of applying the discrete Meyer wavelet, mean, Gaussian, median and Weiner filtering, and non-linear diffusion over four different noise powers are plotted in Table 3.3. Four different noise powers are utilised: $a = 0, 0.5, 1, 1.5$. $a = 0$ corresponds to the case in which no noise is simulated on the input depth image. The denoising algorithms' parameters are selected from the best results reported in [26] and are described in Table. 3.2. For each classification algorithm, the rank-one recognition rate is calculated. Almost for all cases, increasing the noise powers reduces the recognition performance.

When the noise power is low ($a \approx 0$), the KFA face recognition approach has the best performance for all denoising approaches (shown in red in Table 3.3). However, as the noise power increases, median filtering followed by the TreeBagger, m-SVM, KFA, PNN and KNN-Ctb classifiers produce the highest recognition ranks. For instance, for the m-SVM classifier (shown in cyan in Table 3.3), median filtering produces significantly higher rank-one recognition rates compared to the other denoising techniques. A similar trend exists for the other classifiers, as denoted in magenta in the table for the median denoiser.

On the other hand, for the subspace projection-based classification methods, which are PCA, KFA, PNN and LDA, it is the Weiner filtering which generates the highest recognition ranks, when the noise power is high at $a = 1.5$. This is displayed in green in the table. When the noise power is high, the KNN-Ctb and TreeBagger result in the highest rank-one rates. This is denoted in blue for the highly intensive noise power in blue colour ($a = 1.5$), for the KNN-Ctb and TreeBagger in Table 3.3.

To sum up the results, although the performance varies for each classification algorithm, the median and Weiner filters generally result in the highest recognition ranks. Despite their smoothing effects, non-linear diffusion, discrete Meyer wavelet and mean filtering produce the lowest average recognition ranks, which verifies their inability to completely remove the

Algorithm	Noise power	m-SVM	PCA	KFA	PNN	KNN-Ctb	Tree-Bagger	LDA	KNN-Euc	KNN-Cos
Discrete Meyer	$a = 0$	93.83	94.68	97.66	93.62	94.47	93.83	95.53	90.43	84.47
	$a = 0.5$	63.40	62.98	61.70	62.77	85.11	80.00	62.55	62.77	62.98
	$a = 1$	61.91	61.28	61.49	61.70	70.85	71.91	60.85	61.70	61.91
	$a = 1.5$	62.13	61.49	61.91	61.49	68.30	68.94	61.06	61.70	61.49
Mean filtering	$a = 0$	96.38	95.32	99.36	94.04	93.83	93.62	97.23	91.70	84.68
	$a = 0.5$	74.47	75.32	80.00	71.91	89.57	80.09	73.83	70.64	65.32
	$a = 1$	66.38	67.23	71.28	63.62	82.77	77.87	65.53	66.60	57.45
	$a = 1.5$	63.62	64.89	68.51	63.83	73.62	78.09	64.68	62.77	52.55
Gaussian filtering	$a = 0$	95.32	95.11	98.94	92.55	92.98	94.47	97.66	90.00	81.70
	$a = 0.5$	73.62	78.51	83.62	72.34	87.23	84.89	83.40	72.55	69.57
	$a = 1$	67.87	67.45	75.32	65.96	80.64	76.81	70.21	65.74	62.13
	$a = 1.5$	64.04	65.74	72.98	63.19	73.62	73.40	69.15	62.13	59.57
Non-linear diffusion	$a = 0$	94.26	94.26	97.02	92.34	93.83	94.47	92.34	89.79	81.91
	$a = 0.5$	63.62	63.83	63.62	61.91	86.60	82.98	61.28	63.19	63.40
	$a = 1$	61.49	61.28	61.70	61.49	68.94	78.72	60.85	61.28	61.06
	$a = 1.5$	61.06	60.85	61.28	61.49	65.32	75.32	61.06	60.85	61.28
Weiner filtering	$a = 0$	95.32	94.89	98.51	92.34	93.19	91.28	96.81	90.00	81.70
	$a = 0.5$	70.43	75.96	83.40	72.13	86.60	83.62	78.51	70.85	63.40
	$a = 1$	64.26	70.43	77.87	69.15	77.66	74.04	71.28	65.74	57.45
	$a = 1.5$	63.62	68.72	77.45	68.09	74.26	72.77	68.94	65.96	56.60
Median filtering	$a = 0$	95.74	94.47	98.30	92.98	93.40	93.83	96.38	90.85	83.40
	$a = 0.5$	89.15	75.11	85.32	74.68	92.77	90.64	79.79	60.00	46.81
	$a = 1$	80.64	67.23	75.96	69.15	84.89	90.00	72.77	52.77	40.85
	$a = 1.5$	75.11	62.77	72.55	65.96	81.28	89.36	69.15	48.30	35.96

Table 3.3: Rank-one recognition rates (in %) for different noise powers and denoising algorithms. KFA's strength in high performance classification is demonstrated in red, while KNN-Ctb and TreeBagger's robustness against noise in high noise power is denoted in blue. Median filtering's high potential to more successfully denoise faces is signified in magenta and cyan, while the Weiner filtering's higher performance when used prior to subspace projection methods is shown in green.

impulsive spike noise. The lowest classification results are achieved when the cosine and Euclidean distances are applied over the KNN classifier.

Table 3.3 also shows that the performance of the subspace projection methods (PCA, KFA and LDA) significantly deteriorates as the noise level increases. For example, the recognition ranks for the KFA classification method drop by approximately 27% as noise level parameter a is increased from 0 to 1.5. This is because of these methods are very sensitive to the outliers in the data and this is intensified at high noise powers.

In order to have a better view on the comparison of the classification methods performance, a is increased from 0 to 2.75 and the rank-one recognition rates are computed when median filtering is applied as the denoiser. The result, plotted in Fig. 3-11, verifies the results in Table 3.3, showing the KFA and LDA classifiers produce the highest recognition ranks when the noise power is low (≈ 0). However, their performance is drastically decreased for higher noise

power.

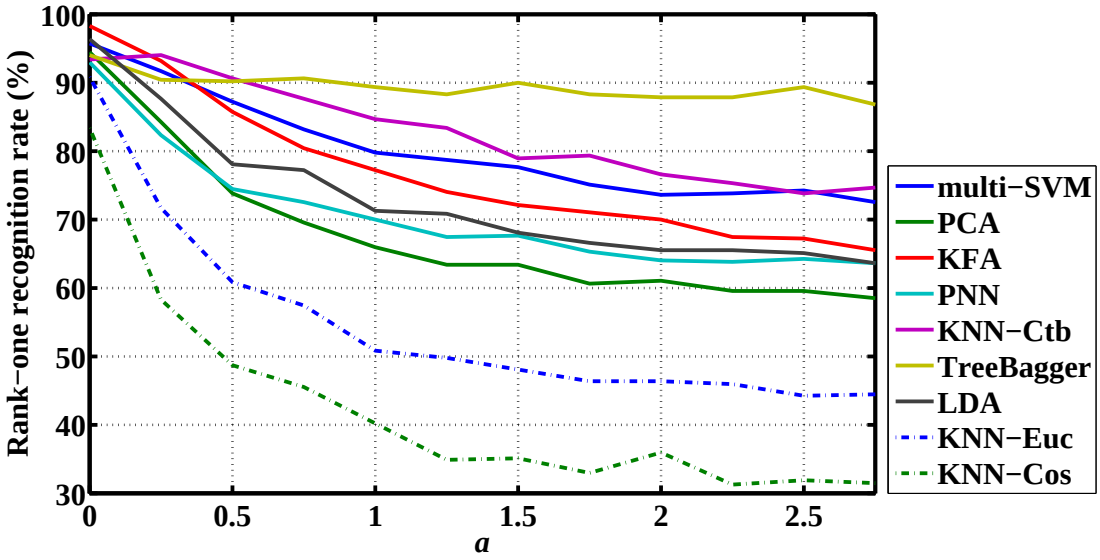


Figure 3-11: Rank-one recognition rates for different noise powers determined by a, when median filtering is applied. As the noise power is increased, the recognition performance of the TreeBagger classifier relatively remains constant, while it declines more rapidly for the other algorithms.

The Tree Bagger classifier is the most robust recognition algorithm, producing the best performance for $a > 0.5$. Its output recognition rate remains very close to 90% even at high values of a. Among the different matching criteria the city-block distance significantly outperforms the Euclidean and cosine distances. This might be due to the higher performance of the L1-norm, when applied on a sparse feature space ([2, 130, 131]).

3.4.7 Noise/denoised gallery vs. Noise/denoised probe

Noise is a stochastic process and randomly changes the depth map. Its distribution might also change for different image acquisition devices. Therefore, the classification algorithm should be robust enough against this variations. As the final experiment, in this section the effects of using noisy and denoised samples for the gallery and probe is evaluated. The purpose of the experiment is to quantify the face recognition algorithms' performance, when the train or test samples are either denoised or degraded by noise.

The result of applying the aforementioned procedure over the Spring 2003 samples is illustrated in Table 3.4 for $a = 0.25$ and median filtering as the denoiser. The four classification methods which produced the highest recognition rates in the previous section are used: TreeBagger, KNN-Ctb, m-SVM and KFA. When a noisy gallery is used, the noisy probe images

are recognised with rank-one recognition rates $\approx 93\%$, with the KNN-Ctb matching algorithm outperforming the training-based methods. Also, as expected, the recognition rates for the case when both the gallery and probe are denoised is high, while the subspace projector KFA classifier produces 98.30% rank-one recognition rate.

Gallery	**Probe**	
	Noisy	**Denoised**
Noisy	TreeBagger: 91.91% KNN-Ctb: 94.04% m-SVM: 91.70% KFA: 93.19%	TreeBagger: 12.55% KNN-Ctb: **45.53%** m-SVM: 1.70% KFA: 10.00%
Denoised	TreeBagger: 15.32% KNN-Ctb: **81.91%** m-SVM: 2.34% KFA: 11.06%	TreeBagger: 93.83% KNN-Ctb: 93.40% m-SVM: 95.74% KFA: 98.30%

Table 3.4: Noisy/denoised gallery vs. Noisy/denoised probe rank-one recognition results, when $a = 0.25$ and median filtering is used for denoising. KNN, which is a matching algorithm, outperforms other classifiers when applied over noisy probe samples to match with denoised gallery images.

This experiment also shows how the classification methods can fail when the samples in the probe (or gallery) have different noise distribution. The classification performance significantly decreases when a noisy gallery is used for denoised probe. A similar situation exists for the case of denoised gallery and noisy probe. Failure in detecting the correct between-class boundaries or subspace projection axes are the main causes of this deterioration. Also, when the probe is noisy and gallery is denoised, the learned classes do not be fit to the unseen noise in the data and under-fitting occurs. As a result, the samples are wrongly classified and recognition ranks decrease. The KNN-Ctb distance, however, still has significantly higher recognition performance than the other leaning-based approaches, as it generates a 81.91% rank-one recognition rate for the denoised gallery when used for the noisy probe samples.

3.5 Conclusion

In this chapter, a study of the effect of different denoising algorithms on some holistic 3D face recognition algorithms is presented. A face recognition pipeline is used, in which the influence of the denoising on all subsequent recognition steps can be evaluated. The denoising

approaches results show that the parameters value used to tune the denoising algorithms can have higher values, and the denoising can confidently be applied more aggressively than those reported in the literature, as this is beneficial to the overall recognition performance. This fact was verified by comparing different classification algorithms on various denoising methods. The quantitative evaluation shows that median filtering with a 33 mask size (which occupies an area of approximately 16.5×16.5 mm) can produce a feature space with which KFA can produce a 98.35% rank-one recognition rate. The proposed method is database independent and can be applied to other types of imaging modalities such as photometric stereo imaging or 4D datasets. Also, the proposed algorithm has the potential to be extended to other biometric approaches. Most biometric system rely on a feature space, which is vulnerable to noise. The recognition-based denoising approach can be applied in order to tune the denoising parameters and choose the best denoising algorithm for a given dataset.

This chapter also explores the robustness of denoising and 3D holistic face recognition algorithms for different noise powers. To be able to quantitatively evaluate the robustness to noise of different classification and denoising methods, the novel approach of learning the noise distribution over the facial surface and then simulating it over other samples is proposed. The 3D face recognition evaluation pipeline is used to evaluate non-linear diffusion, median, mean, Gaussian, Weiner and wavelet filtering as denoising techniques, applied to seven different classification methods, including SVM, neural networks, Tree-based, KNN matching and subspace projection methods.

Median, Gaussian and Weiner filtering generate the best results, with the median filter producing the best overall performance for median to high noise intensity. For low intensity noise, subspace projection classifiers (KFA and LDA) outperform others. However, when the noise intensity is increased, the performance of subspace projection methods significantly deteriorates and, in the experiments, the TreeBagger and KNN with the city-block distance show the best robustness. The use of denoising/noisy samples for the gallery/probe is also evaluated. These result show that matching algorithms (KNN-Ctb) significantly outperforms the training-based methods, as they do not rely on classification boundary allocation or subspace projection.

The proposed method to estimate the noise distribution and applying over faces is completely database independent. One interesting area of future work is to evaluate the performance of the denoising techniques over other types of imaging modalities, such as photometric stereo images [16]. This can help to find suitable denoising methods for use with face datasets employing various imaging modalities. Although this chapter focused on datasets with few training samples, other possible research would be to explore the noise-robustness of other facial recognition algorithms, such as the deep learning-based and sparse classifiers, which rely on higher number of per subject samples. While the results provided in this chapter were

mostly based on identification performance, it would be interesting to explore the verification scenarios performance, when different noisy samples are utilised.

Chapter 4

Nose tip detection

4.1 Introduction

The nose tip is one of the most fundamental landmarks and is usually detected for 2D and 3D face recognition applications. Its location helps to detect and crop the face region, localise appropriate ROIs to detect other facial landmarks and also to segment other facial regions. For the subsequent steps of 3D face recognition algorithms, a consistent and accurate nose tip location is vital step as improperly cropped face can affect the features correspondence and alignment algorithms.

While the algorithms used for 2D nose tip detection are mainly based on using the geometrical shape of the face, for 3D faces it is the curvature information which has been extensively utilised for nose tip detection [67, 40, 2]. The basic idea for these algorithms is that the nose region is highly salient on the facial surface, due to its significant convexity. This can be quantified by thresholding the SI, mean, Gaussian, or principal curvatures. Although the curvature calculation has proven its strength for many applications, such as landmarking and 3D segmentation, it has a number of disadvantages.

1. Curvature calculation relies on differentiation computation on the face region. Differential operators (such as Sobel and Canny filters) are essentially high pass filters. As a consequence, they also amplify the high frequency noise components as well.

2. Although the nasal region's convexity is highly remarkable on the face, there are still other convex regions, which might be misclassified after curvature calculation. In other words, other regions such as the forehead, chin, and sometimes cheeks or lips might have

very close curvature characteristic to the nose tip area. Moreover, in a facial biometric data acquisition system, there are sometimes redundant captured points, usually related to the neck, chest and shoulders. These regions may also include some convex parts such that it can be very difficult, if not impossible, to find consistent thresholds for the curvature functions to detect the nose tip region in a large dataset.

3. Although curvature calculation is roll rotation invariant, it is sensitive to yaw and pitch rotations.

A new algorithm for nose tip detection, which is significantly less sensitive to curvature thresholds, self- and partial occlusion and noise, is proposed in this chapter. It is mainly based on the nasal region's salient convexity quantified by filling the inverted nose. Instead of preallocated fixed thresholds, an increasing thresholding band is defined over the SI and candidates for the nose tip are detected as centroids of the convex regions. Then, for each point a sphere is centralised and intersected with the face surface.

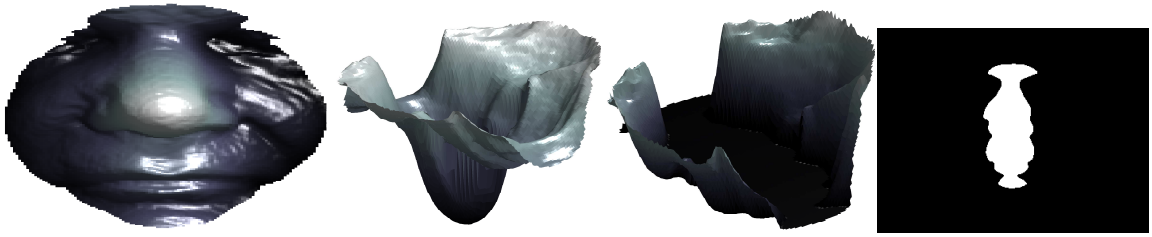
The regions are cropped and the integral of the filled inverted nose is stored for each point. Since the nasal region is significantly more consistent than other convex facial regions and produces a large consistent inverted filling result, a histogram of the integral images produces a dominant peak for the nose tip region. In order to make the algorithm robust against possible rotations around the pitch and yaw directions, the face is rotated around the x and y axes and the same procedure is repeated at each rotation step. All the candidate points are eventually clustered using a Gaussian heat map and the peak is selected as the indicator of the best nose tip.

4.2 Overview of the inverted nose filling algorithm

The nose region's continuous convexity can be detected using a method named "the inverted filling algorithm" (IFill). The basic premise behind the algorithm is that if the nose region's depth is inverted (Fig. 4-1-b) and its morphological filling calculated (Fig. 4-1-c), as the nasal region is convex, it will produce a continuous connected component. Detecting the largest connected region in the filled image results in a binary image ($\mathbf{B}$). Therefore, IFill acts as an operator, which maps the input 3D face's depth ($\mathbf{Z}$) in $\mathbb{R}$ domain, shown in Fig. 4-1-a, which is in $\mathbb{R}$ domain, into the binary domain $\{0, 1\}$, Fig. 4-1-d. This operation is

$$\mathbf{B} = \text{IFill}(\mathbf{Z}), \tag{4.1}$$

in which $\mathbf{B}$ and $\mathbf{Z}$ have equal size.



(a) (b) (c) (d)

Figure 4-1: (a) the cropped face region; (b) The inverted nasal region; (c) The filled image; (d) The largest connected region in the filled image.

4.3 Improved thresholding of convex regions using thresholding bands

Thresholding the SI map, using predefined thresholds is sensitive to the noise. Since it is a bottom-up approach for the nose tip detection, prior knowledge of the nose shape is not integrated and, therefore, other convex regions present in the input point clouds may be detected as the nose tip. Instead of preallocating values to threshold SI, a thresholding band is defined and iteratively increased around a central SI value. SI is then thresholded and the largest connected region is detected. To be more specific, the following operation is performed over the SI map,

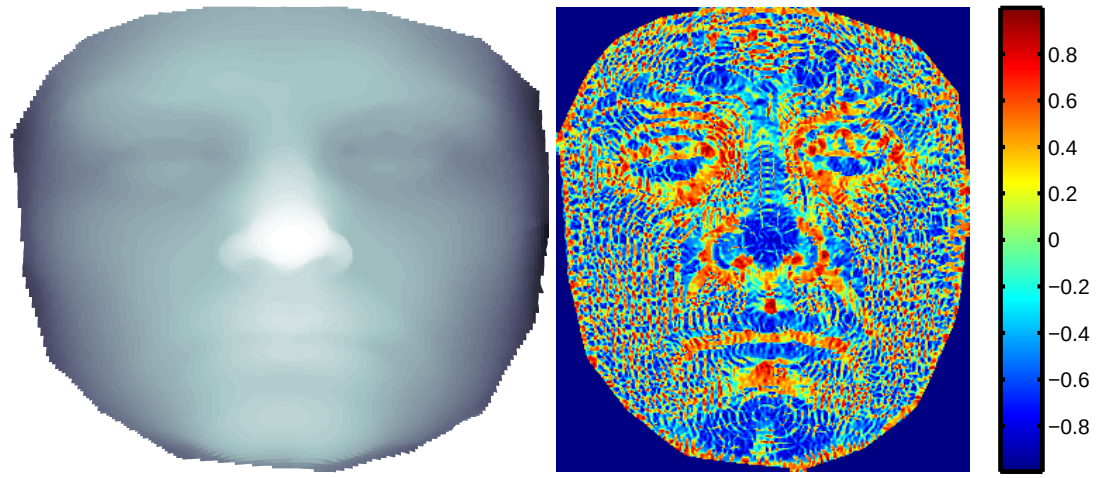
$$\begin{cases} \mathbf{B}_{ij} = \mathbf{L}_j(\text{SI}, T_i) \\ \text{SI} = \frac{2}{\pi} \arctan\left(\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}\right) \end{cases} \tag{4.2}$$

where $\mathbf{L}_j(.)$ finds the j^{th} largest connected component ($j = 1, \ldots, N$) of the thresholded SI, using the i^{th} thresholding band T_i ($i = 1, \ldots, M$). $\mathbf{B}_{ij}$ is a binary image, containing the j^{th} largest connected convex region computed by the i^{th} thresholding band. T_i is localised around a central SI's value (in this work $-\frac{6}{8}$) and is increased in each iteration, based on the recursive equation,

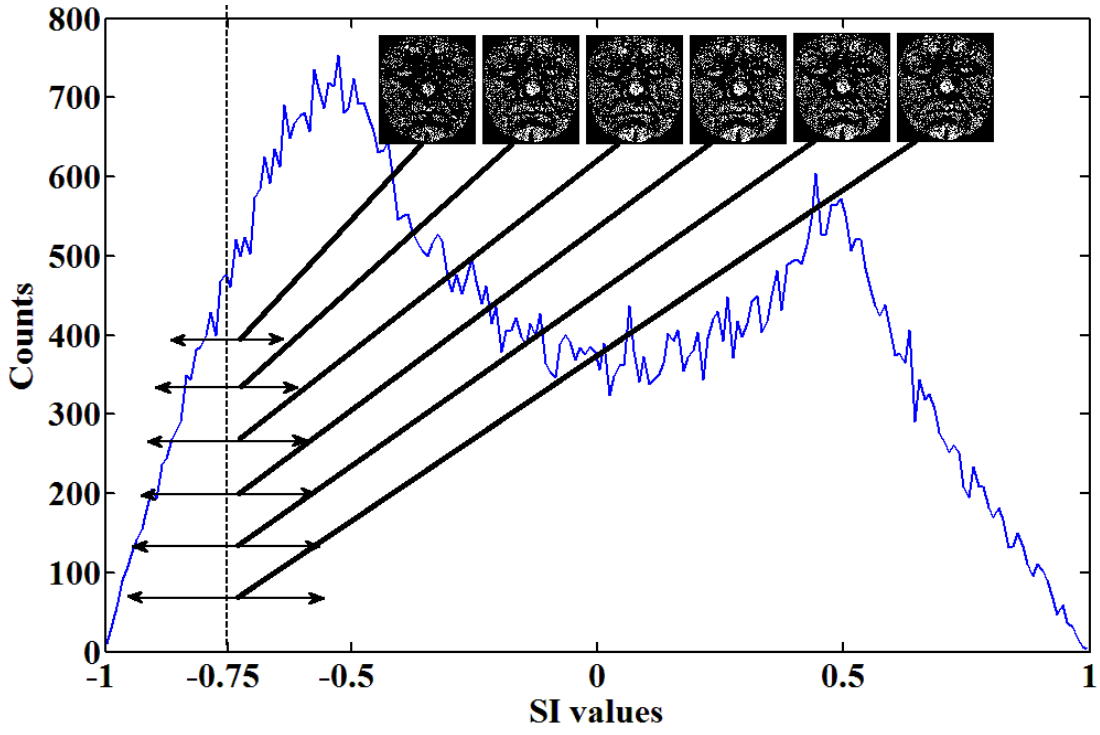
$$T_i = T_{i-1} + \frac{1}{a^i}, \tag{4.3}$$

a is a scalar (> 1) used to determine the increase rate of T_i. As i is increased, the increment rate of T_i reduces exponentially, which helps to emphasize the higher thresholding bands. An example of the effects of choosing different values for T_i on the thresholded image is shown in Fig. 4-2. Figure 4-2-c shows the thresholding results after using six thresholding bands, around the central SI -0.75. For each thresholded image, $\mathbf{B}_{ij}$ is computed.

The result of computing $\mathbf{B}_{ij}$ for three thresholding bands and the seven largest connected



(a) (b)

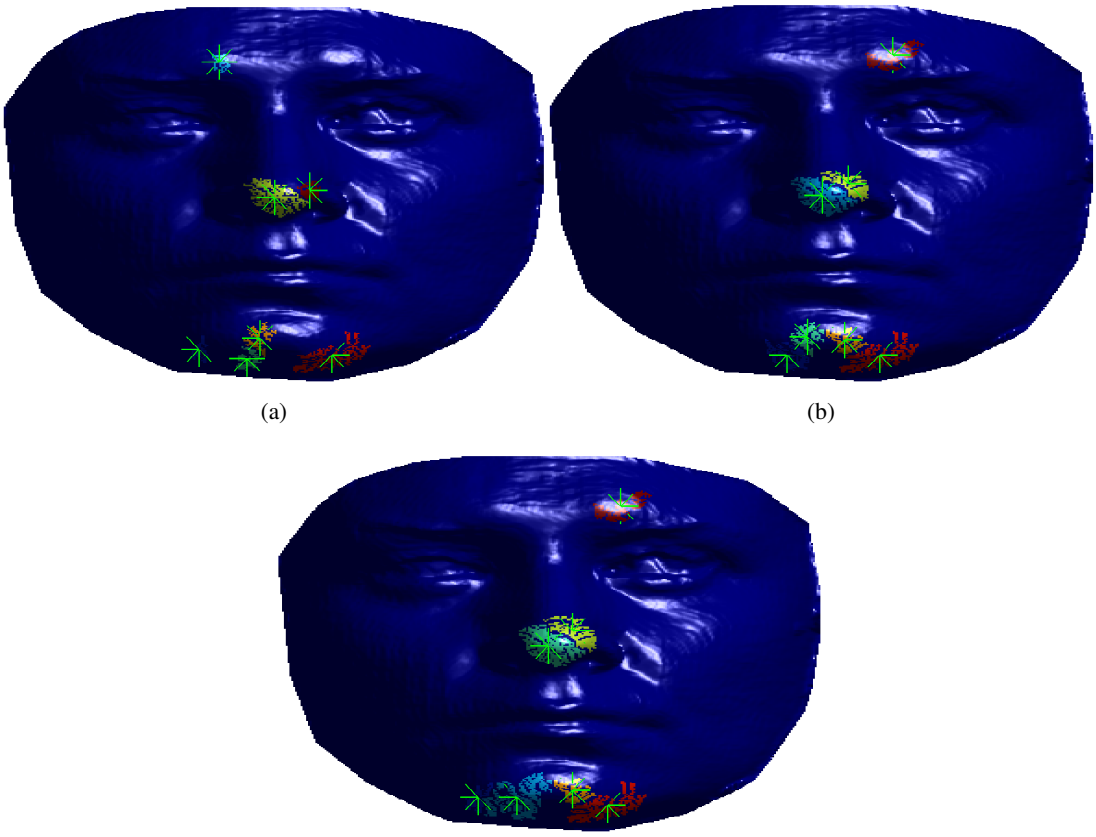


(c)

Figure 4-2: (a) An example facial depth image; (b) SI map; (c) Convex maps using different thresholding bands over SI.

components in a typical convex map ($j = \{1, 2, \ldots, 7\}$) are depicted in Fig. 4-3. The centroid of each $\mathbf{B}_{ij}$ is plotted in green points.

A sphere centered at $\mathbf{B}_{ij}$'s centroid ($\mathbf{C}_{ij}$), with radius $65mm$, is then intersected with the



(a) (b)

(c)

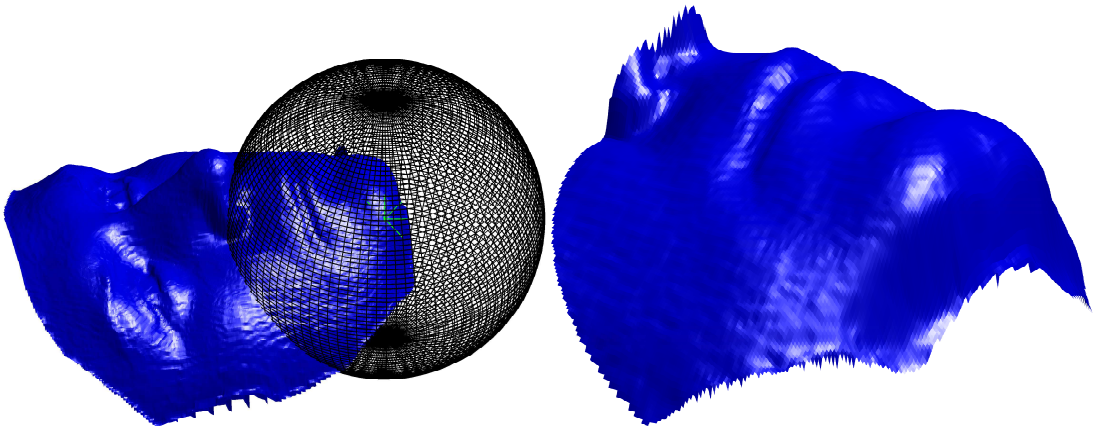
Figure 4-3: $\mathbf{B}_{ij}$ for the thresholding bands: (a) $T_1 = 0.1250$, (b) $T_2 = 0.1563$ and (c) $T_9 = 0.1667$.

input 3D surface of the face and its inner part is cropped, as shown in Fig. 4-4 for one of the centroids.

Then, the IFill(.) operator is applied to the cropped region's depth map ($\mathbf{Z}_{ij}$). In order to decrease the computational time of the IFill(.) operator, the largest connected region detection is avoided. Instead, the region, in which $\mathbf{C}_{ij}$ is located, is detected as $\mathbf{B}_{ij}^{I}$,

$$\mathbf{B}_{ij}^{I} = \text{IFill}(\mathbf{Z}_{ij}, \mathbf{C}_{ij}). \tag{4.4}$$

The result is plotted in Fig. 4-5. This simple modification in IFill(.)'s computation not only increases the computational speed in every iteration but also keeps $\mathbf{C}_{ij}$ closer to possible locations of the nose tip. To be more specific, the largest connected region in the filled image might not necessarily correspond to the nasal region. This might occur when there are several large convex regions in the cropped image. As a result, the largest connected region in the



(a) (b)

Figure 4-4: (a) The sphere intersection with the face surface resulting the cropped region in (b).

filled image might not necessarily indicate the nose region. Using $\mathbf{C}_{ij}$ in the IFill(.) operator significantly helps to overcome this issue.

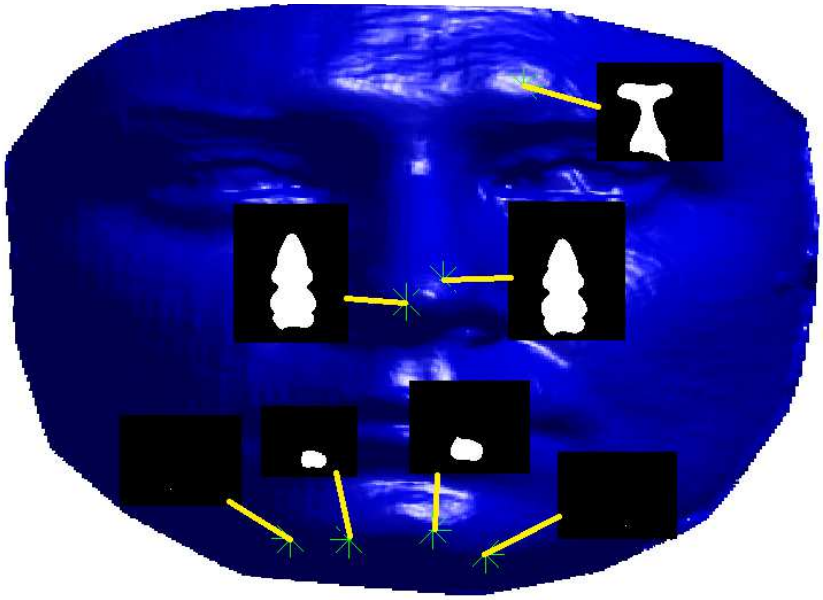
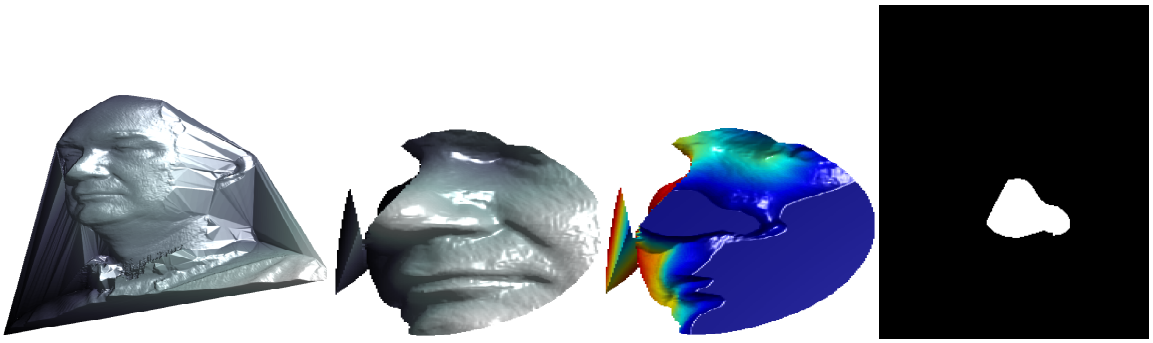


Figure 4-5: $\mathbf{B}_{ij}^{I}$ computed using $\mathbf{C}_{ij}$ ($j = \{1, 2, \ldots, 7\}$).

Moreover, defining IFill using (4.4) helps to apply the inverted filling algorithm over the self-occluded samples as well. For these cases, the invalid points on the boundaries are replaced by the median value of the valid points and a similar procedure is then applied, i.e the depth map is inverted and filled by the morphological filling algorithm. The result of this operation

is plotted in Fig. 4-6 for a self-occluded face.



(a) (b) (c) (d)

Figure 4-6: (a) A self-occluded face image; (b) the cropped region using the spherical intersection; (c) the filled inverted depth image; (d) the IFill(.) operator result.

After computing IFill over each cropped surface, the sum of all foreground pixels in the binary image is computed by

$$E_{ij} = \sum_{x,y} \mathbf{B}^I_{ij}. \tag{4.5}$$

4.4 Obtaining the candidate points

The nasal region's salient convexity causes E_{ij} to have repeated values at different thresholding level i. Therefore, if E_{ij} is lexicographically ordered and its histogram is found, the P highest peaks correspond to the most frequented regions, one of which will correspond to the nose region. Figure 4-7 shows the histogram of E_{ij}. The three peaks marked in the histogram correspond to the most frequent regions of $\mathbf{B}^I_{ij}$ and, as was expected, one of peaks corresponds to the nasal region.

4.5 The nose tip detection procedure

The input 3D face captures will typically be at random rotations along the yaw and pitch directions. This might degrade the nasal region's convexity saliency and other parts, such as the forehead, chin, cheeks or hair, produce larger $\mathbf{B}_{ij}$. Also, due to the pose variations, self-occlusion will occur and significant parts of the nose might be lost. This might reduce the frequency of those E_{ij} related to the nasal region in the E_{ij} histogram and, as a result, the nose tip will not be detected using the highest peaks of the histogram.

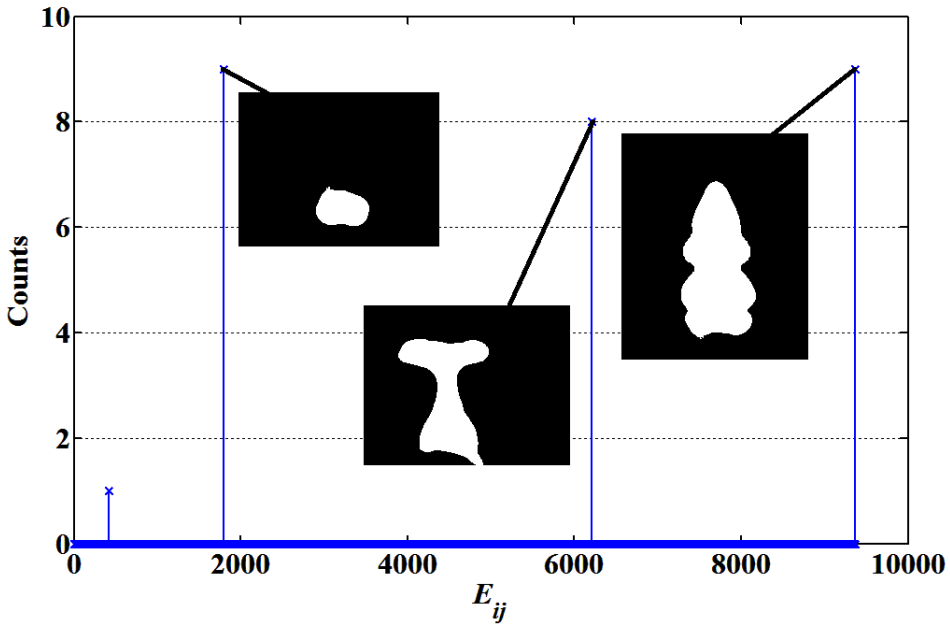


Figure 4-7: The histogram of E_{ij}. $\mathbf{B}_{ij}^{I}$ of its corresponding three highest peaks are also plotted. The peaks occurred at $E_{ij} \approx 9200$ relates to the nasal region, while the other peaks correspond to other convex regions on the face.

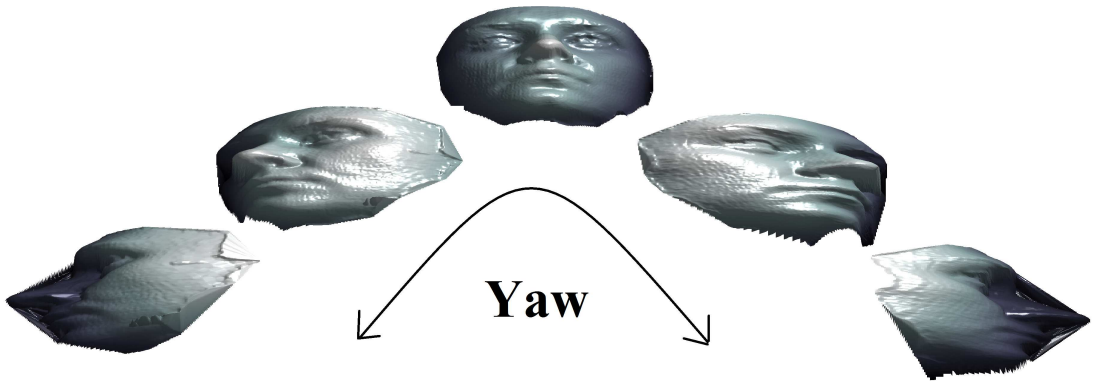
In order to resolve the problems caused by pose variations, the face is rotated along the yaw and pitch directions, as shown in Fig. 4-8-a and -b. The points which become self-occluded at each iteration are removed and, finally, the face is resampled using a uniform grid. In each angular rotation, the algorithm used to compute E_{ij} is performed and the P highest peaks in the resulting histograms are detected.

All of the candidates points for the nose tip, gathered from different rotations, are then remapped onto the original face coordinates (Fig. 4-9-a). A 2D Gaussian function centralised over each point is then defined,

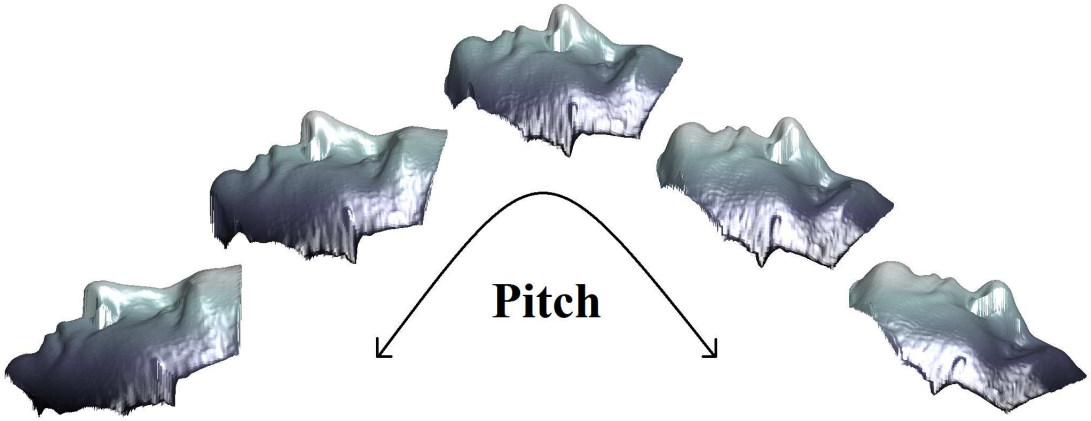
$$\mathbf{G}_k = A \exp(-\frac{(\mathbf{X} - T_x^k)^2}{2\sigma_x^2} - \frac{(\mathbf{Y} - T_y^k)^2}{2\sigma_y^2}). \tag{4.6}$$

$\mathbf{G}_k$ is the Gaussian function computed over the k^{th} nose tip candidate, $\mathbf{T}^k = [T_x^k, T_y^k, T_z^k]$ $(k = 1, 2, \ldots, K)$. $\mathbf{X}$ and $\mathbf{Y}$ are the input face's coordinate maps. $\sigma_x = \sigma_y = \sigma$ are the standard deviation of the Gaussian function and A is its amplitude. The Gaussian functions are the accumulated to create a heat map ($\mathbf{H}$),

$$\mathbf{H} = \sum_{k=1}^{K} \mathbf{G}_k, \tag{4.7}$$



(a)

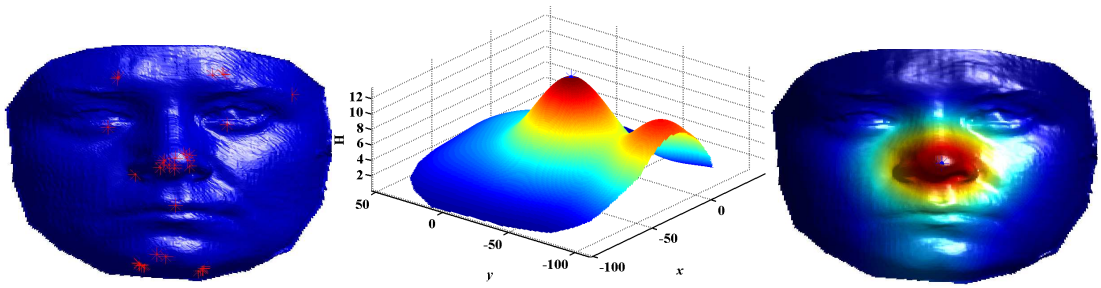


(b)

Figure 4-8: The convention used to define (a) yaw and (b) pitch rotations over 3D faces.

in which K is number of detected points as candidate for the nose tip. $\mathbf{T}^k$ will be highly dense around the nose tip location and causes a significant increase in the heat map. Therefore, calculating $\mathbf{H}$'s maximum corresponds to the nose tip location. $\mathbf{H}$ is depicted in Fig. 4-9-b, in which its maximum is detected. Also, $\mathbf{H}$ is simultaneously plotted over the 3D face surface in Fig. 4-9-c. As it is shown, $\mathbf{H}$'s maximum corresponds to the nose tip.

The proposed algorithm can be presented in a compact form as shown in Algorithm 1. The input point clouds are rotated along the pitch and yaw directions, in the range of $[-\theta_p^{\min}, \theta_p^{\max}]$ and $[-\theta_y^{\min}, \theta_y^{\max}]$, respectively. At each rotated angle, the face is resampled using Delaunay triangulation and a uniform resolution grid. Then the SI map is computed. The i^{th} thresholding band is used to find the j largest connected components in the thresholded SI map. For each connected region, the centroid $\mathbf{C}_{ij}$ is detected. Then, a sphere is centralised over $\mathbf{C}_{ij}$ and $\mathbf{B}_{ij}^I$ is computed for the cropped region. The P highest peaks of the $\mathbf{B}_{ij}^I$ integrals are localised. Repeating this procedure for all the pitch and yaw rotations results in a set of points for the nose tip ($\mathbf{T}^k$) and are used in (4.6) to find the heat map. The maximum of the heat map corresponds



(a) (b) (c)

Figure 4-9: (a) All the points as candidate for the nose tip remapped from different yaw and pitch directions; (b) The heat map (**H**); (c) The heat map plotted over the 3D face image.

to the nose tip.

4.6 Experimental results

The nose tip detection algorithm is tested on the Bosphorus dataset, which includes different intensive facial expression types and self-occluded samples. The self-occluded samples have pose variations along the yaw axis, which caused significant data loss on one side of the face. As the ground truth values of angular rotations are provided, the dataset provides the capability to determine the range of poses over which the algorithm is able to successfully detect the nose tip. The effects of varying the algorithm's parameters are also quantitatively evaluated on a subset of the Bosphorus dataset.

4.6.1 Algorithm's parameters analysis

The proposed algorithm for the nose tip detection includes a set of tunable parameters. Four of the most important of them are: 1 and 2) yaw and pitch angle increment steps, which are used to compute θ_p and θ_y at every iteration; 3) N, the number of largest connected components of the thresholded SI map; and 4) the number of P highest peaks in the histogram of E_{ij}. Different values for these parameters are used and a set of K candidate points for the nose tip are detected. Then, the distance of the closest point to the ground truth is computed. Finally for all samples in the dataset, the average and standard deviation of the samples' error are calculated.

The purpose of this experiment is to check, how successful the algorithm is in finding the nose tip location, if the above parameters are varied, prior to the Gaussian functions calculation. The results are depicted in Fig. 4-10, 4-11 and 4-12. For all experiments, $\mathbf{D}_{min}$ is the vector

Algorithm 1 The nose tip detector

procedure NTD($\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$)
  for $\theta_p = -\theta_p^{\min}$ **to** $\theta_p^{\max}$ **do** ▷ Do for a set of given pitch angles
    for $\theta_y = -\theta_y^{\min}$ **to** $\theta_y^{\max}$ **do** ▷ Do for a set of given yaw angles
      $[\mathbf{X}_{py}, \mathbf{Y}_{py}, \mathbf{Z}_{py}] \leftarrow$ ROTATE($\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$, θ_p, θ_y)
      $[\mathbf{X}_{py}, \mathbf{Y}_{py}, \mathbf{Z}_{py}] \leftarrow$ APPLYSELFOCC($\mathbf{X}_{py}$, $\mathbf{Y}_{py}$, $\mathbf{Z}_{py}$) ▷ Removes the occluded points
      $[\mathbf{X}_{py}, \mathbf{Y}_{py}, \mathbf{Z}_{py}] \leftarrow$ RESAMPLE($\mathbf{X}_{py}$, $\mathbf{Y}_{py}$, $\mathbf{Z}_{py}$)
      SI = CURVCALC($\mathbf{X}_{py}$, $\mathbf{Y}_{py}$, $\mathbf{Z}_{py}$) ▷ Finds the SI
      for $i = 1$ **to** M **do**
        for $j = 1$ **to** N **do**
          $T_i \leftarrow T_{i-1} + \frac{1}{a^i}$ ▷ Allocates the new thresholding band
          $\mathbf{B}_{ij}$ = $\mathbf{L}_j$(**SI**, T_i)
          $\mathbf{C}_{ij}$ = FINDCENTROID($\mathbf{B}_{ij}$) ▷ Computes the centroid
          $\mathbf{Z}_{ij}$ = CROP($\mathbf{X}_{py}^r$, $\mathbf{Y}_{py}^r$, $\mathbf{Z}_{py}^r$, $\mathbf{C}_{ij}$) ▷ Crops the depth map, using the current centroid
          $\mathbf{B}_{ij}^I$ = IFILL($\mathbf{Z}_{ij}, \mathbf{C}_{ij}$)
          E_{ij} = $\sum_{x,y} \mathbf{B}_{ij}^I$ ▷ Finds the binary image's integral
        end for
      end for
      $\mathbf{T}^k \leftarrow$ FINDPEAKS1D(HIST(E_{ij}), P) ▷ Finds the P highest peaks of the 1D E_{ij}'s histogram
      $\mathbf{T}^k \leftarrow$ REMAP($\mathbf{T}^k$, θ_p, θ_y) ▷ Relocates the tip candidate points to the original coordinates
    end for
  end for
  for $k = 1$ **to** K **do**
    $\mathbf{G}_k \leftarrow \mathbf{G}_k(\mathbf{T}^k, \sigma)$ ▷ Computes the Gaussian function using (4.6)
  end for
  H = $\sum \mathbf{G}_k$ ▷ Computes the heat map (4.7)
  TIP = FINDPEAKS2D(**H**, 1) ▷ Find the 1st highest peak (the maximum) from the 2D heat map
  return TIP
end procedure

whose elements correspond to the minimum Euclidean distances of the nose tip candidates to the ground truth per dataset sample. When $\mathbf{D}_{min}$ is computed for all samples, its mean and standard deviation can be calculated and plotted for different parameters.

The first experiment is performed over different values for N, see Fig. 4-10. As N increases, the probability of having the nose tip in one of the N largest connected components, in the convex map increases. Therefore, although for $N = 1$ the error is approximately $12mm$, it drastically decreases for higher values for N and converges to around $2.66mm$ for $N \geq 3$. This shows that when $N = 1$, the largest connected region calculated by thresholding SI does not necessarily correspond to the nose region (although this procedure has been used extensively for the nose region detection by other authors [67, 40, 2, 44]). This is because there are some other convex regions located on the face, such as forehead, cheeks, and lips, which can produce larger convex regions when the SI is thresholded. On the other hand, when $N \geq 3$ regions are found, it is significantly more probable that one of the N regions corresponds to the nose tip.

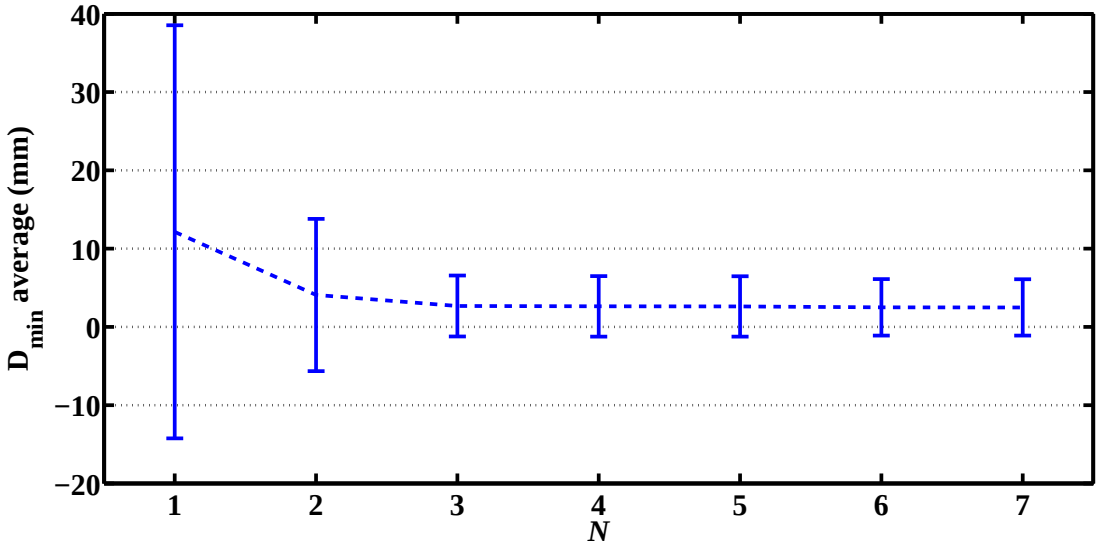


Figure 4-10: The average of $\mathbf{D}_{min}$ for different values of N, when $P = 10$ and yaw and pitch angular increment steps are 15°.

Another experiment is performed for different values of P, see Fig. 4-12. When P increases, more peaks are detected from E_{ij}'s histogram. As a consequence, increasing P will result in a higher probability that the nose tip is among the set of candidate points for the nose tip ($\mathbf{T}^k$). For $P = 1$, only one peak is selected from the histogram for each angular rotation. As a consequence, the error is approximately $7.5mm$. However, the error decreases significantly when $P \geq 2$ and plateaus at $\approx 2.7mm$.

Both the experiments show that the algorithm performs better for larger values of P and

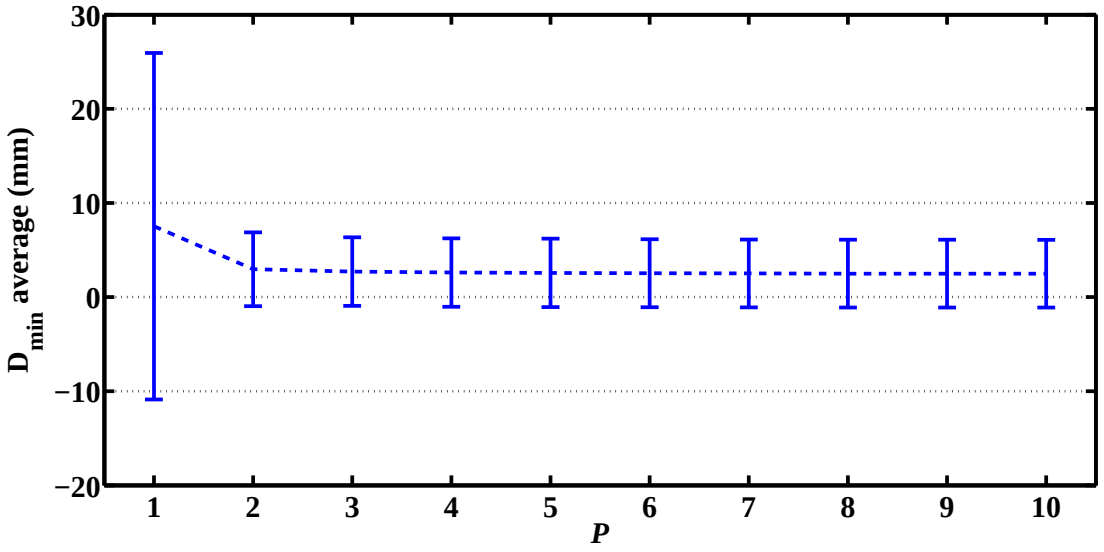


Figure 4-11: The average of $\mathbf{D}_{min}$ for different values of P, when $N = 7$ and yaw and pitch angular increment steps are 15°.

N. Although increasing P and N results in higher computational cost, they both converge when $N \geq 3$ or $P \geq 2$. This shows that the proposed nose tip detector becomes insensitive to parameters change after relatively low thresholds and so, lower values can be confidently chosen for P and N, without having a high detection error.

The final experiment in this section is to evaluate the effects of varying the angular increment steps used to compute θ_p and θ_y at each iteration. This is tested for three steps: 5°, 10°and 15°. For all the angular steps, $\theta_p^{\max} = -\theta_p^{\min} = 15°$and $\theta_y^{\max} = -\theta_y^{\min} = 30°$, and the results applied over the frontal view samples of 15 subjects in the Bosphorus dataset is shown in Fig. 4-12. Lower angular increment steps would slightly reduce the output error. This is because the candidates for the nose tip ($\mathbf{T}^k$) will be found more densely around the nose tip area and therefore, $\mathbf{D}_{min}$'s average error reduces. However, the increase in the error is not that significant as the angular increment step is increase (about $0.3mm$ from 5°to 15°). Reducing the angular step significantly increases the computational cost. The overall number of pitch and yaw rotations are 72 for 5°and 8 for 15°increment steps, which shows how much the computation increases for lower angular steps. However, as is shown by this experiment, lower resolutions can be used to assign values to θ_p and θ_y at each iteration, without increasing the risk of high output error.

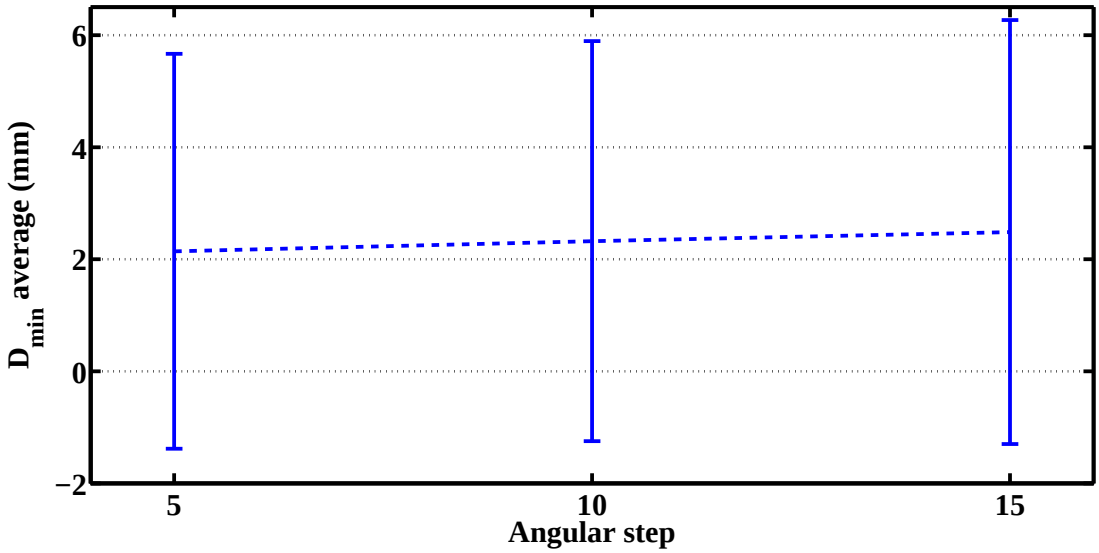


Figure 4-12: The average and standard deviation of $\mathbf{D}_{min}$ for different values of angular increment step, when $N = 4$ and $P = 5$, applied over the frontal view samples of 15 subjects in the Bosphorus dataset. As the angular step increases the accuracy of the nose tip detection slightly decreases.

4.6.2 Precision curve and thresholded distance to the ground truth

Precision curves, which are standard methods to quantify the accuracy of landmarking algorithms [151] are used in this section to evaluate the landmarking results. The locations of the nose tip found by the proposed algorithm are compared with the ground truth. This comparison can be quantified by the distances to the ground truth, which are the manually localised nose tip locations provided by the Bosphorus dataset, vs. the average of the number of samples whose nose tip location have been correctly detected within the given distance. The precision curve is computed over 2920 samples (from 105 subjects) in the Bosphorus dataset with frontal views, but extensive expression variations.

The curves are plotted for two cases: when $\sigma = 5mm$ and $\sigma = 30mm$. σ is used to create the heat map after finding the addition of the Gaussian functions. When σ is large, the resulting heat map has wider peaks and as a consequence, the location of the nose tip (which corresponds to the highest peak location), is less accurate. This is demonstrated in Fig. 4-14-a. Alternatively, when σ is smaller, the peaks are much sharper and the nose tip's location becomes more accurate (Fig. 4-14-b). This can be seen in Fig. 4-13, where $\sigma = 5mm$ has resulted in lower distances from the ground truth compared to $\sigma = 30mm$. It should be mentioned that if σ is too small, the Gaussian maps generated by the set of $\mathbf{T}^k$ cluster close to the nose tip area will not be merged properly and this deteriorates the nose tip detection accuracy.

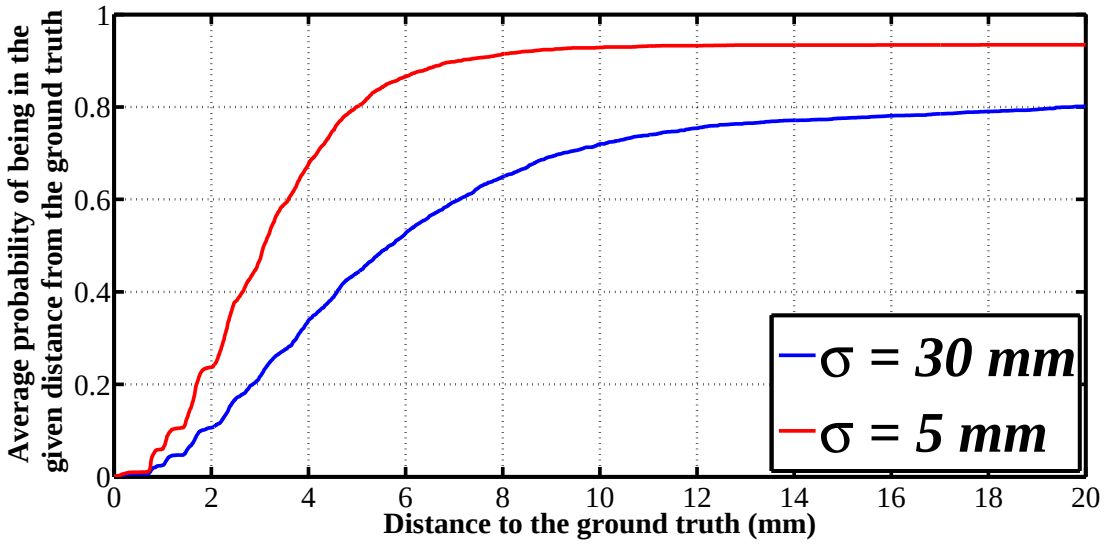
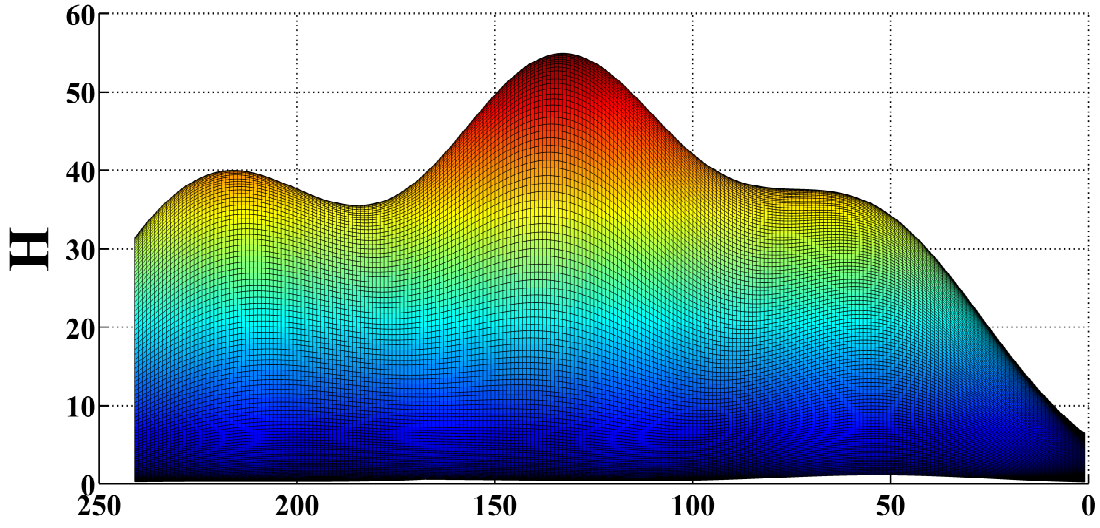


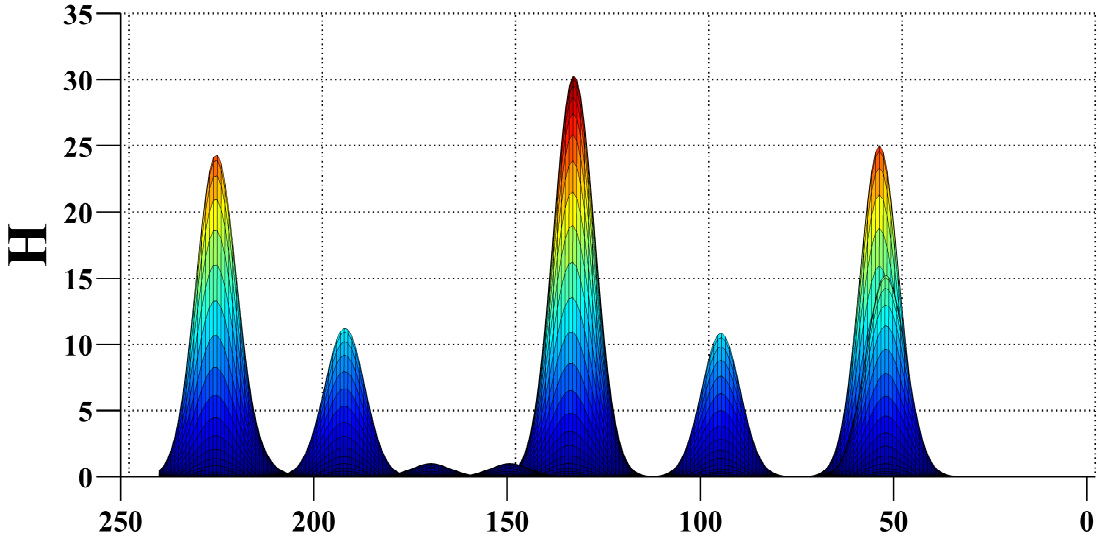
Figure 4-13: Precision curves for the nose tip detector, when the standard deviation of the Gaussian maps used for the accumulative heat map are different. Lower σ produces sharper peaks in the Gaussian maps, resulting in more accurate landmarking.

The Bosphorus dataset includes samples rotated along the yaw by specific angles. The accuracy results achieved by the proposed nose tip compared with the corresponding ground truth for these samples are demonstrated in Table. 4.1. These are the average errors for a nose tip to be detected within the range of $7mm$ distance from the ground truth. The amount of yaw directional rotated faces are 10°(R10), 20°(R20), 45°(R45 and L45) and 90°(R90 and L90), where R and L represent rotations to the right and left, respectively. The experiment is performed over 105 samples for each type of rotations, for the 105 different subjects in the Bosphorus dataset.

The algorithm can detect the nose tip within the range of $7mm$ distance for the L45, R45, R30 and R20 with 100% accuracy. But surprisingly, its performance fails to 91.7% when R10 samples are used. This is because for L45, R45, R30 and R20, some parts of the face, which might have been wrongly detected as the nose tip due to its convexity, have been removed by the self-occlusion and this data loss is much less for the R10 samples. The algorithm's performance for the R90 and L90 samples significantly deteriorates, because nearly half of the facial region is lost because of the self-occlusion. As a result, some other parts such as the hair or ears are detected as the most convex parts and wrongly classified as the nose tip.



(a)



(b)

Figure 4-14: Side view of the heat maps for: (a) $\sigma = 30mm$ and (b) $\sigma = 5mm$. The more sharpness in the highest peak in (b) results in a more accurate nose tip localisation than (a), which is also shown in Fig. 4-13.

4.7 Conclusion

This chapter has introduced an algorithm for, in general, the nasal region, and in particular, nose tip detection. The algorithm is based on novel binary signature definition over the nose tip region, which maps the points from the real to the binary domain. Instead of assigning predefined thresholds, a thresholding band is incrementally increased and a set of points are detected as the nasal tip region at each step. These points are used to create an energy function,

Rotation degrees	**L45**	**L90**	**R10**	**R20**	**R30**	**R45**	**R90**
Detection accuracy	100%	25%	91.7%	100%	100%	100%	33.3%

Table 4.1: The nose tip detector's accuracy over the Bosphorus dataset yaw occluded samples.

whose histogram peaks discriminate with the most potential points to be the nose tip.

In order to overcome the misclassification of other convex facial parts, such as the forehead, hair, lips and chin, the above procedure is performed over different pitch and yaw rotated faces. Finally, the found points are used to create a heat map, whose maximum indicates the nose tip location. The algorithm was tested on the Bosphorus dataset and reaches $\approx 90\%$ correct recognition in the range of $7mm$ over various facial expressions. Also, the algorithm is robust against yaw self-occlusion, until at least 45°rotations. In summary, the algorithm provides the capability to overcome the issues involved with the previous direct curvature thresholding and is able to detect the nose tip from different expressions and self-occluded faces.

The major issue with the proposed method is its computational complexity. This is mainly because of the need to calculate the morphological filling at each rotation. Since the rotations applied using θ_p and θ_y and the operations at each rotations are completely independent, one way to solve the computation issue might be to parallelise these steps. Then the set of points as candidates for the nose tip can be concatenated afterwards, to compute the heat map.

Chapter 5

Face alignment using the nasal region

5.1 Introduction

Three-dimensional image alignment is the process of rotating, scaling and translating an image, in order to represent the object of interest in a known posture, which is a vital step in many object recognition algorithms.

In most of the 3D face recognition approaches, alignment is a vital step in order to obtain a successful recognition. Inaccurate localisation caused by an inconsistent alignment can result in poor feature sets which in turn affect the feature space distribution and the recognition performance. To address this problem numerous 3D face alignment algorithms have been proposed, that can be broadly categorised as either register-based or self-dependent.

To address these problems, this chapter proposes a novel self-dependent algorithm for 3D face rotational alignment that uses the nose region. Compared to other parts of the face, the nose is one of most stable regions over different expressions [67, 40] and is also relatively easy to detect. The proposed approach only requires an approximate position of the nose tip as its landmark and so avoids the need for sophisticated landmark detection, one of the common problems of self-dependent algorithms. After finding a rectangle enclosing the nose region the algorithm defines the nose footprint as the largest filled region in the inverted depth map. An energy function, defined for this footprint, is minimised by rotating the face around the x- and y-axes. Finally, alignment is performed for the z-axis by iteratively applying a function minimisation to the face symmetry map until a stop condition is satisfied.

The remainder of this chapter is organised as follows. In Section 5.2 the stages of the proposed method, termed the nose region self-dependent 3D face rotational alignment algo-

rithm, are explained in more detail. An experimental evaluation is performed in Section 5.3 and discussion and conclusions are given in Section 5.4.

5.2 Nose Region-Based Self-dependent 3D Face Rotational Alignment

The main stages of the algorithm are shown in Fig. 5-1. The pre-processing step applies a fuzzy *C*-means (FCM) outlier removal algorithm [149] to classify the range data as pixels inside or outside the body. Outliers within the body are then replaced by the median of their surrounding body pixels. Morphological filling is used to fill any residual holes in the depth map and two applications of a 5×5 median filter applied to remove any spikes.

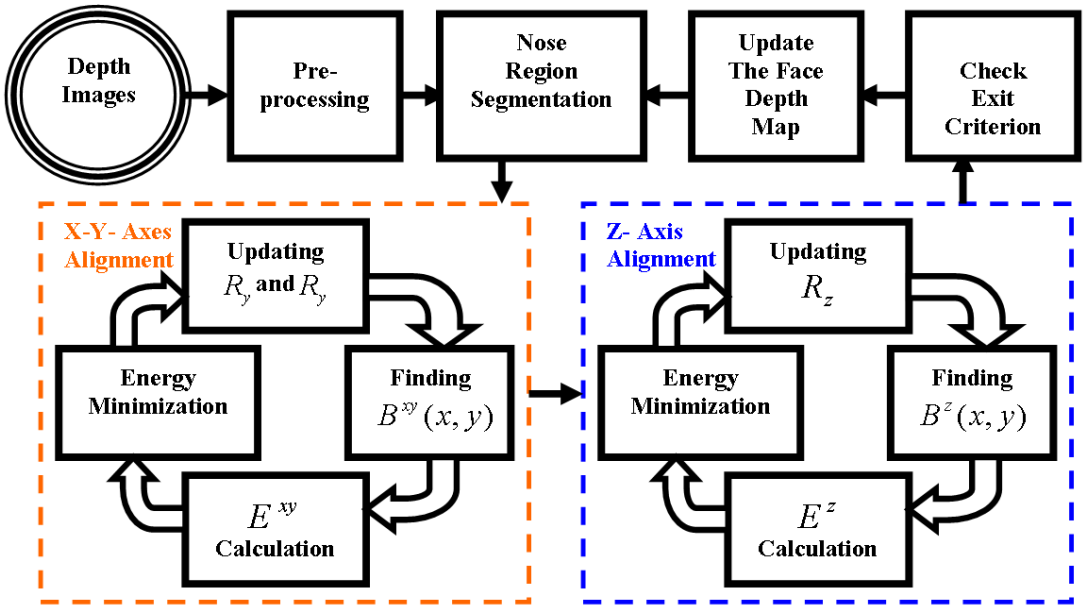


Figure 5-1: Block diagram of the proposed alignment method. R_x, R_y and R_z are the rotation matrices around the x-, y- and z-axes, respectively.

Using the nose tip, the nose region is then approximately localised on the face. For an aligned 3D face the nose tip is usually the closest point to the camera. It is also one of the largest convex regions on the depth map. Although an aligned image is not initially available, the algorithm uses candidate nose tips locations provided by the curvature of the face region to iteratively update the nose tip location using the function minimisation approaches described below. After detecting the nose tip information on the spatial resolution of the images is used

to define a fixed sized rectangle that localises the nose. An example rectangle for an image from the FRGC dataset is shown in Fig. 5-2.

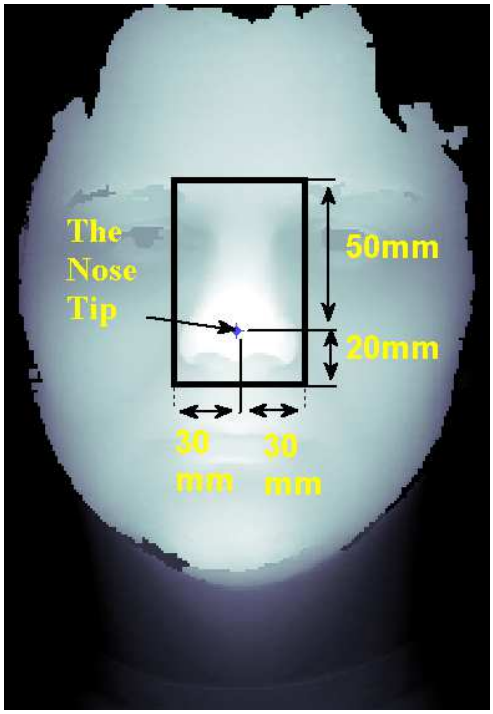
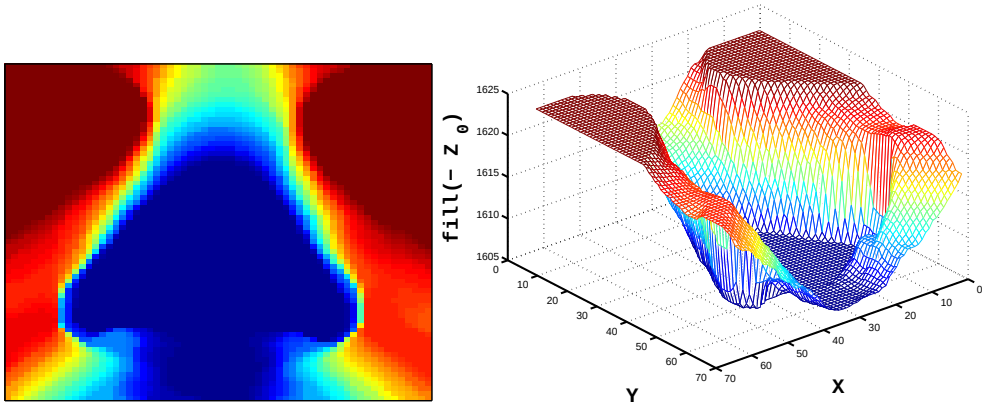


Figure 5-2: The rectangle used for nose segmentation.

5.2.1 Filled Depth Map Updating by Energy Minimisation

In this section, the core of the proposed algorithm is explained, which is based on the inverted nose filling algorithm demonstrated in the previous chapter, as shown in Fig. 5-3. The nose footprint is then identified as the largest connected component in the filled image and Fig. 5-4 presents a binary image of this region.

Maximising the area occupied by the connected region in Fig. 5-3-a will remove the rotation around the x- and y- axes. In other words, if the sum of black pixels in the binary map is maximised, the range image would be positioned so that the maximum projection of the filled image is made on the xy plane. By rotating the image so that the line of symmetry (shown in Fig. 5-4) is parallel with the y-axis, the rotation around the z-axis will be eliminated. Since the area calculation in the binary map is z-axis rotation invariant, alignment is first performed for the x- and y-axes and then the alignment procedure continues on the resulting depth map for z-axis alignment. These two alignment procedures are described in more detail below.



(a) 2D (b) 3D

Figure 5-3: The filled image of the nose region. The blue regions represent a flat connected component.

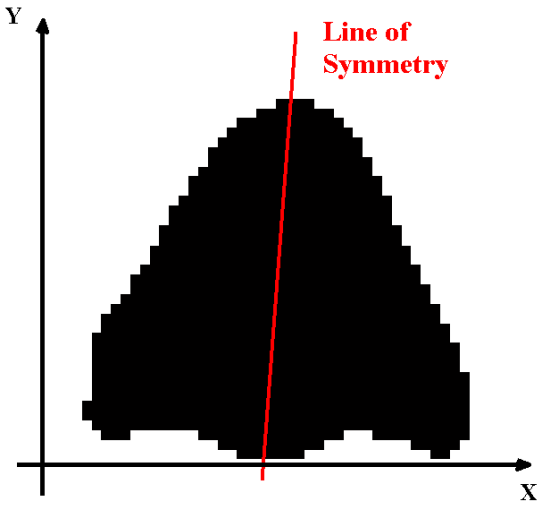


Figure 5-4: The largest connected component from the filled image of Fig. 5-3.

Finally, a convergence criterion is evaluated. Here, the procedure is repeated until

$$\sum |[\theta^{\mathbf{min}}]_{t+1} - [\theta^{\mathbf{min}}]_t| < T \tag{5.1}$$

where T is a threshold set by the user and $[\theta^{\mathbf{min}}]_t$ and $[\theta^{\mathbf{min}}]_{t+1}$ are 1×3 vectors, containing the x, y and z rotational angles at iterations t and $t+1$, respectively. This condition is similar to the situation when the energy functions' outputs become approximately flat in the evolutionary computation.

5.2.2 x- and y-axes Rotational Alignment

During this step the segmented nose region is rotated iteratively in the x- and y-axes in order to minimise an energy function. Let IFill($\mathbf{Z}$) denote a nonlinear operation which fills the inverted depth map and extracts the biggest connected region,

$$\begin{cases} \mathbf{B^{xy}} = \text{IFill}(\mathbf{Z}) \\ \mathbf{Z} \in \Re \rightarrow \mathbf{B^{xy}} \in \{\, 0,\, 1 \,\} \end{cases} \tag{5.2}$$

where $\mathbf{B^{xy}}$ is the output binary image. The energy function used for this section is

$$E^{xy} = -\log\left(\sum_{i,j} \mathbf{B^{xy}}(x_i, y_j)\right), \quad \begin{matrix} i = 1, \cdots, M \\ j = 1, \cdots, N \end{matrix} \tag{5.3}$$

where *M* and *N* are the number of columns and rows in $\mathbf{B^{xy}}$, respectively.

Finally, simulated annealing (SA) is used for energy minimisation. As a global optimisation method, SA is relatively insensitive to initialisation and, in many applications, has been found to be faster than pattern search and genetic algorithm (GA) [62, 57]. SA is applied for a fixed number of iterations and then the evolution is continued from the lowest value of the energy function using a local optimisation method. In this work, the Levenberg–Marqurdt (LM) algorithm is used to locally minimise the energy function [152].

5.2.3 z-axis Rotational Alignment

After removing the rotations around the x- and y-axes, rotation removal around z-axis is performed. Similar to the previous section, $\mathbf{B^z}$ is computed using (5.2). The energy function used here is based on the *y*-axis symmetry of $\mathbf{B^z}$, see Fig. 5-4. To calculate this symmetry, in every iteration $\mathbf{B^z}$ is divided into two parts, using the nose tip location previously found. The new location of the nose tip is also mapped by utilising the rotation matrices.

Next, the two halves of $\mathbf{B^z}$ are compared and the number of unequal pixels found. Consequently, E^z, the energy function for *z*-axis rotation alignment will be,

$$\begin{aligned} E^z = \sum_{i,j} \mathbf{XOR}\left(\mathbf{B^z_{\text{Left}}}\left(x_i, y_j\right), \mathbf{B^z_{Right}}\left(x_i, y_j\right)\right), \\ i = 1, \ldots, M', \quad j = 1, \ldots, N' \end{aligned} \tag{5.4}$$

where M' and N' are the number of columns and rows in each half, respectively, and **XOR**$(\cdot)$ is the logical Exclusive OR operator. When necessary, zero padding is used to ensure that both $\mathbf{B}^{\mathbf{z}}_{\mathbf{Left}}$ and $\mathbf{B}^{\mathbf{z}}_{\mathbf{Right}}$ (left and right half images) are the same size. SA is used for evolving the energy function. Since only one variable is updated in each iteration, the SA evolution rate is acceptable for *z*-axis rotation alignment. The point corresponding to the minimum value of E^z, i.e. $\theta_z^{\min}$, gives the rotation required to make the image symmetric along the *y*-axis. In other words, $\theta_z^{\min}$ indicates the degrees that the face had been rotated around the *z*-axis. The depth map is then updated using the value of $\theta_z^{\min}$.

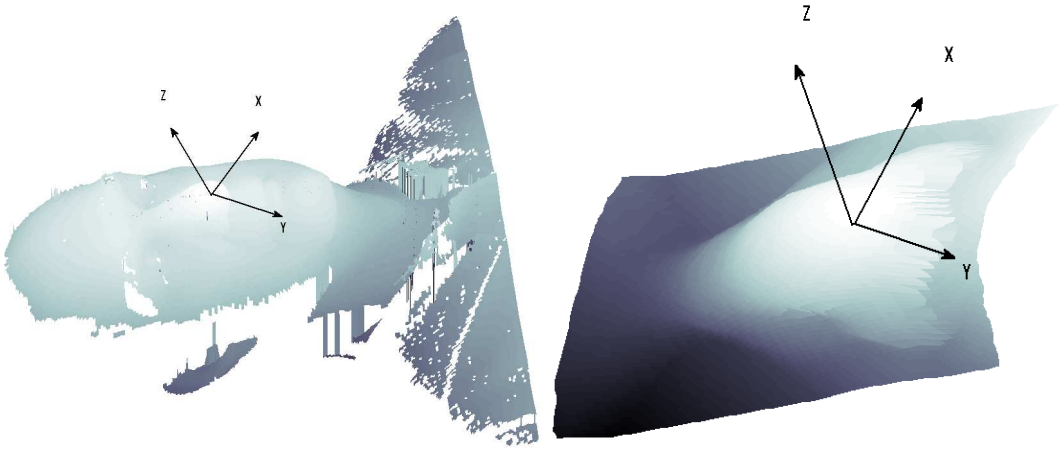
5.3 Experimental Evaluation

The performance of the proposed alignment algorithm is first investigated in application to an example image from the FRGC v2 dataset [14]. A quantitative evaluation of the alignment performance is then undertaken using the ICP algorithm as a comparison technique for the 557 subjects with multiple images in the FRGC dataset. Finally, some qualitative results are presented. In all experiments, 100 iterations of the SA algorithm followed by 125 iterations of the LM algorithm were used for *x*- and *y*-axes rotational alignment and 250 iterations of the SA algorithm for the *z*-axis alignment.

Figure 5-5-a shows an example range image from the "Spring 2003" folder in the FRGC dataset that has been manually rotated in order to evaluate the proposed algorithm. After the preprocessing and nose region segmentation steps described in Section 5.2, the image plotted in Fig. 5-5-b results. To validate the alignment approach, the image in Fig. 5-5-a was rotated around all three axes by $\theta_x = \theta_y = \theta_z|_{t=0} = \frac{\pi}{3}$. As described in Section 5.2.2, the filled area $\mathbf{B}^{\mathbf{xy}}$ needs to be maximised in order to remove rotations around the *x*- and *y*-axes. This procedure is performed by minimising the energy function in (5.3). Figure 5-6-a shows the initial state for $\mathbf{B}^{\mathbf{xy}}$, as calculated from the initialisation of Fig. 5-5-b. Due to the rotations around the *x*- and *y* axes, Fig. 5-6-a shows that initially only a small region is found for the filled binary image. However, as the energy function is minimised the size of $\mathbf{B}^{\mathbf{xy}}$ increases, to give the $\mathbf{B}^{\mathbf{xy}}$ images corresponding to the 65^{th} and final iterations in Fig. 5-6-b and -c, respectively.

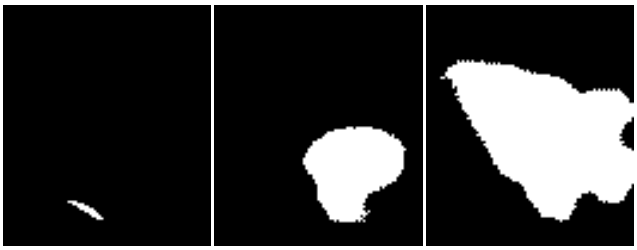
By using the optimum values for θ_x and θ_y, the depth map can easily be updated. In order to remove the rotation around the *z*-axis, the algorithm finds the symmetry line on the updated range image and then makes it parallel to the *y*-axis. In other words, the binary map resulting from (5.2), plotted in Fig. 5-6-c, should be rotated in such a way that the energy function in (5.4) becomes minimised.

The alignment result after both minimisations is shown in Fig. 5-7-d. Also, $\mathbf{B}^{\mathbf{z}}_{\mathbf{Left}}$, $\mathbf{B}^{\mathbf{z}}_{\mathbf{Right}}$



(a) Unaligned image (b) Segmented nose region

Figure 5-5: Example nose segmentation for FRGC image 02463d452

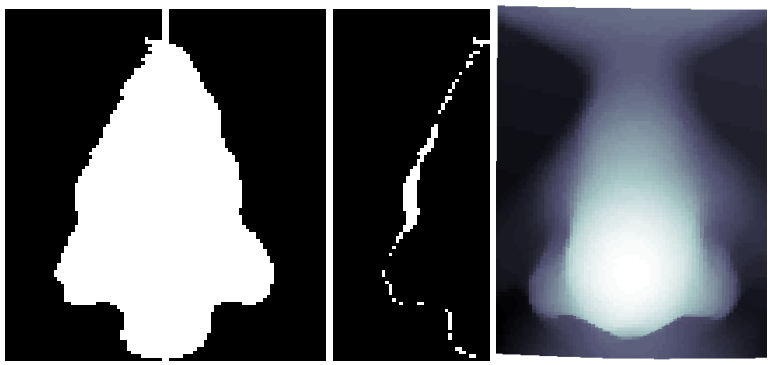


(a) (b) (c)

Figure 5-6: Binary image $\mathbf{B^{xy}}$ at the (a) 1^{st}, (b) 65^{th} and (c) final iterations.

and their logical XOR are shown in Fig. 5-7-a, -b, and -c, respectively (Obviously, $\mathbf{B^{z}_{Right}}$ must be folded in each iteration in order to get the result in Fig. 5-7-c).

Figure 5-8 shows the value of the energy functions through the iterations. For E^{xy} (Fig. 5-8-a) the minimum value over the 100 SA iterations is used as the starting point for the LM minimisation and it can be seen that the LM evolution converges well from this point. For the E^{z} minimisation the performance of SA with the single variable, z, converges well from iteration 120 onwards. For this particular example, the nose tip has been found accurately at the first step and the algorithm immediately moves on to performing the minimisations. These steps continue until the exit criterion, given by (5.1) with $T = 0.01°$, is satisfied. However, for more complex situations, the nose tip is not correctly located properly in the first stage. For such cases, the alignment procedure is repeated until the exit criterion holds.



(a) (b) (c) (d)

Figure 5-7: (a) $\mathbf{B}^{\mathbf{z}}_{\mathbf{Left}}$, (b) $\mathbf{B}^{\mathbf{z}}_{\mathbf{Right}}$, (c) their Exclusive OR and (d) the final aligned nose.

5.3.1 Performance Evaluation on FRGC Dataset

Here, the performance of the proposed alignment method is evaluated using the ICP algorithm as a reference technique. As ICP is a register-based algorithm it requires a reference image to align the test images with. As such, it can not be directly compared to the proposed self-dependent nose region alignment technique. However, using the approach detailed below, ICP can be used to evaluate the alignments achieved by the proposed approach. First, all of the images for the subjects with multiple captures in the FRGC dataset were aligned using the proposed algorithm. Then, for each subject, one image at a time was considered as the reference image and the remaining $N - 1$ images were aligned using the ICP algorithm. For this experiment an implementation of the k-d tree algorithm was used for the ICP point search [56]. For each reference and test image pair, the rotation matrix calculated by ICP is used to measure quality of the alignment. If the reference and test images are perfectly aligned using the proposed method, the resulting ICP rotation matrix will be a 3×3 identity matrix and hence calculating the Euclidean distance between the ICP rotation matrix and the identity matrix provides a quantitative measure of the alignment accuracy. For each subject, the median Euclidean distance for the $N-1$ images was used as a representative error measure. Figure 5-9 shows the median Euclidean distances for all 557 subject with multiple images in the FRGC dataset.

The average of these alignment errors is 0.0315 ± 0.0759 and Fig. 5-10 shows the average rotation matrix, found by averaging the rotation matrices that gave the median error for each subject. Examination of the average rotation matrix's elements shows that it is very close to the identity matrix. To provide a visual interpretation of the alignment error, Fig. 5-13 shows 7 aligned images for one subject, ID 04727, from the dataset. For well registered images with Gaussian noise the surfaces should frequently cross creating a "splotchy" surface [153]. Fig. 5-

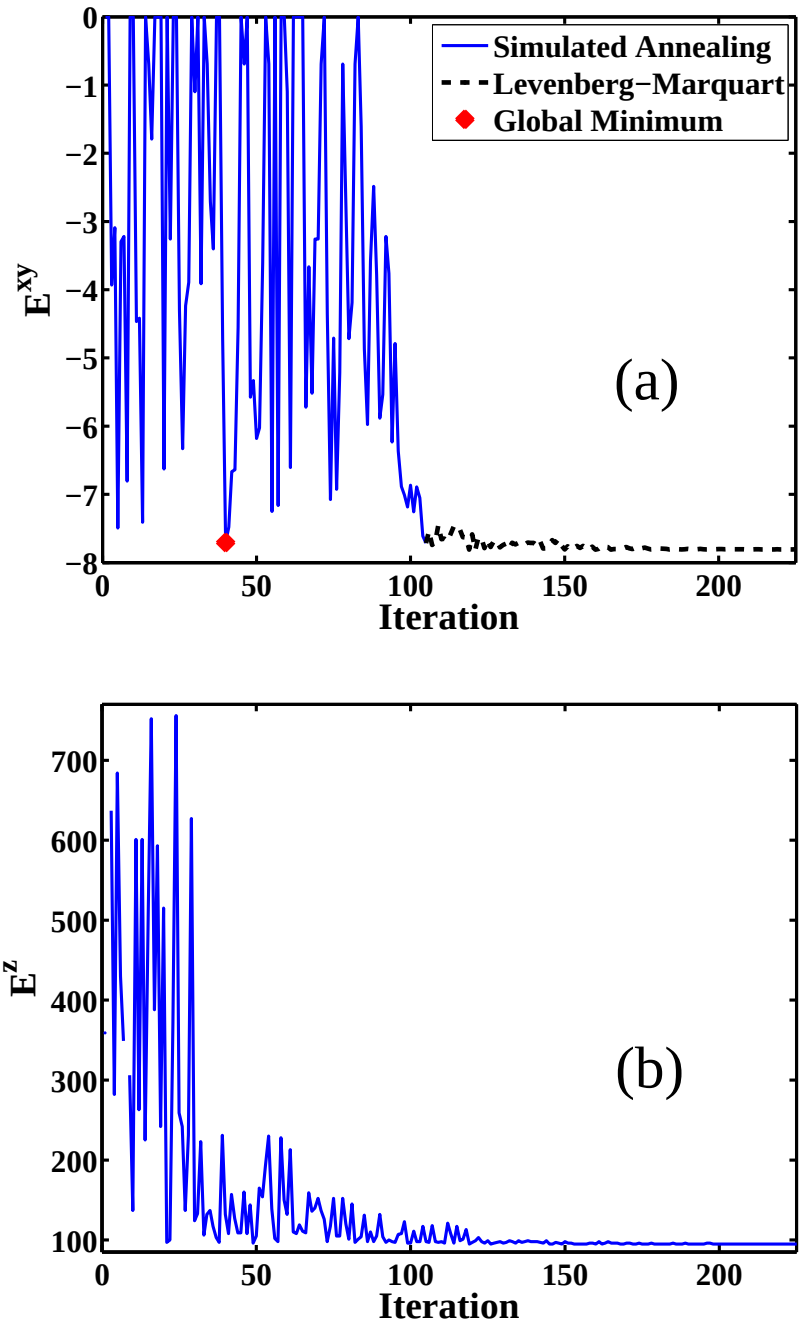


Figure 5-8: (a) E^{xy} and (b) E^{z} evolutions.

13 shows a splotchy surface indicative of good alignment despite the fact that the alignment error for this subject of 0.7011 is significantly worse than average.

To provide a visual illustration of the intra-class similarity for the aligned images, the

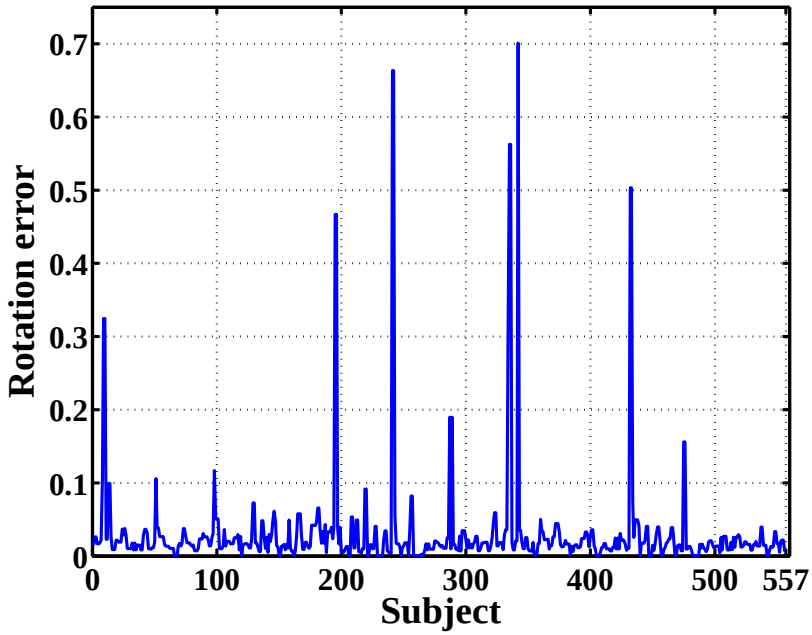


Figure 5-9: Alignment error, measured in comparison to ICP alignment, for all 557 subjects in the FRGC dataset.

$$\begin{bmatrix} 0.9545 \pm 0.1914 & 0.0005 \pm 0.0428 & 0.0031 \pm 0.0497 \\ -0.0003 \pm 0.0483 & 0.9930 \pm 0.0508 & 0.0034 \pm 0.0266 \\ -0.0033 \pm 0.0424 & 0.0023 \pm 0.0124 & 0.9512 \pm 0.2097 \end{bmatrix}$$

Figure 5-10: Average ICP rotation matrix for 557 subjects in FRGC dataset $\pm$ 1 std dev.

original images and final binary maps for 4 different subjects are plotted in Fig. 5-11. These subjects were selected as there is significant variation of expression over their sets of images. However, despite these variations the resulting binary maps of the nose region show consistent alignment and in all cases the algorithm successfully converged.

Finally, the run-time of the Matlab implementation of the proposed self-dependent alignment algorithm over all the samples in the FRGC dataset is shown in Fig. 5-12. The average and standard deviation is 2.1671 $\pm$ 0.5773s. In comparison, using one image per subject as the reference image, the Matlab implementation of a brute force ICP algorithm: `http://www.csse.uwa.edu.au/~ajmal/code/icp2.m`
took an average of 6.9064 $\pm$ 3.1815 to converge. Although faster implementations exist for both approaches, the Matlab run-time provides an approximate, albeit limited, measure of the relative complexity in the absence of a a more accurate complexity analysis.

Figure 5-11: The binary maps for different sessions; Subject ID: (a) 04430, (b) 04916, (c) 04743 and (d) 04851.

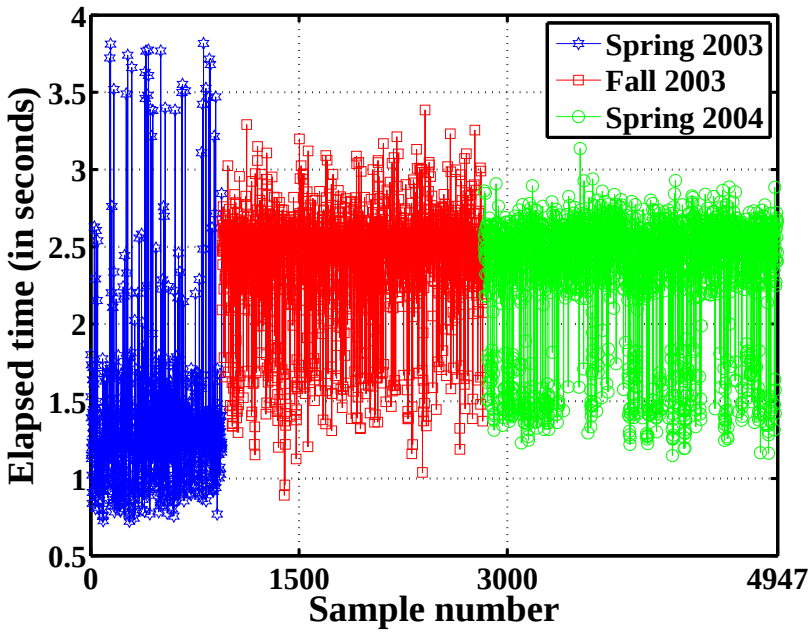


Figure 5-12: Elapsed time for aligning each sample in the FRGC dataset.

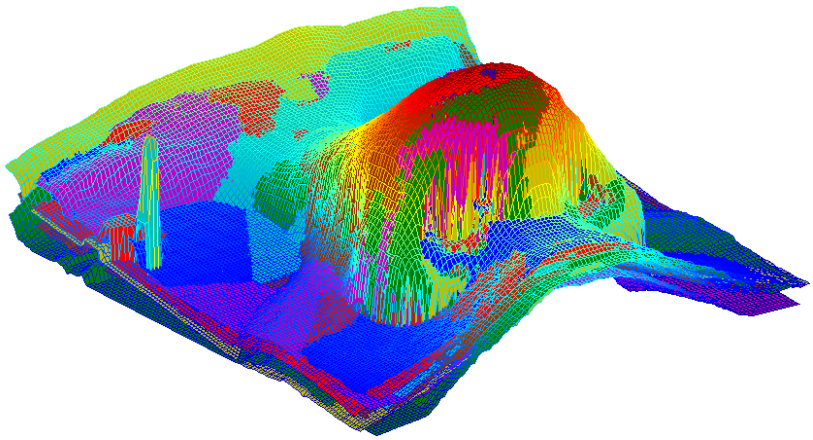


Figure 5-13: Aligned depth maps for FRGC subject 04727.

5.4 Discussion and Conclusions

In this chapter, a novel algorithm is proposed for 3D face rotational alignment using the nasal region. Using the nasal region is advantageous as it is relatively unaffected by variations in expression. The alignment procedure is performed on the nose footprint, which is defined as the largest filled region in the inverted range image. Two energy functions are defined on the nose footprint, which are then minimised by optimisation methods. The first of these is minimised using SA followed by the LM algorithm and removes the rotations around the x- and y-axes. In the second minimisation SA is applied to remove to rotations around the z-axis. After convergence of the energy functions, the rotations around the x-, y-, and z-axes required for alignment are found.

The new method is a self-alignment approach and therefore, unlike register-based algorithms, does not require a reference image and registration algorithm. In addition, by only localising the nose tip, which is one of the most salient facial landmarks, it avoids the need for sophisticated landmarks detection that is a feature of many current self-alignment methods. Both of these features contribute significantly to the low computational complexity that is achieved for 3D face alignment. As the approach is independent of image resolution, image down-sampling has the potential to significantly increase the processing speed and this is an area of future investigation.

Using the FRGC dataset, the new alignment algorithm has been compared to ICP. Results show that the rotational alignment error between the proposed approach and ICP is very small. In addition, unlike ICP which uses the minimum square error as its cost function and can become trapped in local minima, by virtue of its novel energy functions the proposed algorithm evolves through local minima and only required a single run to achieve all its alignment results.

Although the pre-processing step used was specific to the FRGC dataset, the other algo-

rithm stages are generic and can therefore be applied to any arbitrary 3D face dataset. Investigating the alignment performance with photometric stereo face images is an area of further work. The final binary map $\mathbf{B^z}$, in its final evolution iteration, can also be interpreted as a segmentation map for the nose and can be utilised to explore its use in investigations into the suitability of the nose region as a biometric.

Chapter 6

Occlusion-robust facial alignment in 3D

6.1 Introduction

This chapter proposes a novel shape-based algorithm for 3D face pose correction. Similar to the previous chapter, the heart of the new algorithm is the mapping of the 3D cloud points from the nose region to a 2D binary image, termed the nose footprint. The face is then aligned by rotating the nose region around the x, y and z axes such that the largest and most symmetrical nose footprint results. This is achieved by iteratively minimizing a novel objective function on the binary nose footprint. The objective function considers both the symmetry and the area of the nose footprint and has a distinctive global minimum, corresponding to the correctly aligned pose of the face. The optimisation is performed using a pyramidal SA algorithm.

The resulting face pose correction technique is robust to self-occlusion and missing data, and preserves the within-class consistency. It can also be performed at different scales. Compared to the initial version that was presented in the previous chapter, the new algorithm contains an improved objective function and a new pyramidal SA algorithm. Further, while the approach of the previous chapter only considered images from the FRGC dataset [14], the new approach is evaluated on three different 3D face datasets (FRGC [14], UMB [15] and the Bosphorus dataset [13]) and also presents results for simulated self-occlusion.

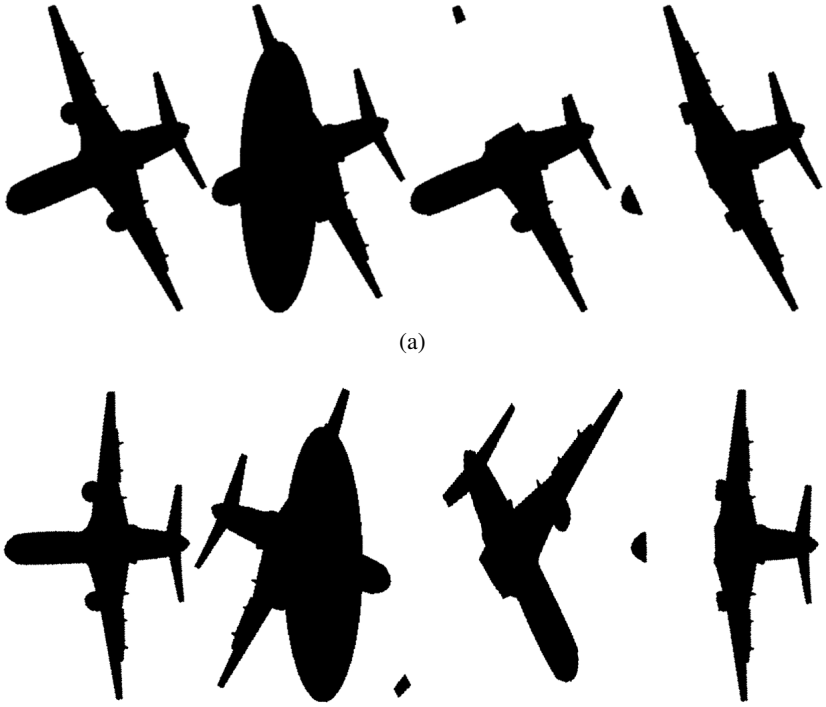
Although PCA based face alignment is computationally efficient, is very appropriate for

coarse alignment and can very successfully correct the roll rotations, it has some noticeable disadvantages. Firstly for the successful alignment to be achieved, sufficient 3D facial points must be available. In other words, although it is robust to limited or symmetrical data loss (such as the self-occlusion due to the pitch rotation) if the data loss is more significant it can fail completely. Problems can occur if some parts of the face are lost because of the image acquisition problems such as noise, self-occlusion (in particular in the yaw direction), and deformations due to intense expression variation. In addition problems arise if the capture has redundant points due to occlusion or incorrect cropping. The presence of any residual impulsive noise or outliers can also significantly degrade the alignment result. The other major problem with PCA is that the data requires post-processing. This is due to the possibility that the order of x, y and z can be changed when the 3D points are mapped to the principal axes. In addition, unwanted rotations and inversions of the facial data can occur, which can be very difficult to detect. These PCA-based alignment issues are plotted in Fig. 6-1 for an example 2D image. In the figure, a binary plane image is rotated, occluded and some of its parts are removed to simulate missing data, which can be caused by an improper denoising or self-occlusion in real 3D imaging. PCA can successfully align the figure when it has been rotated or contains symmetrically missing data. However, it fails when the data is occluded or unsymmetrical. In these cases, due to the distortion, the principal axes found are different and the points are mapped to different locations.

Section 6.2 introduces two operators to be applied on the nasal region. The use of these operators to define an objective function for alignment and the optimisation approach using pyramidal SA are then explained in section 6.3. In section 6.4 the recognition performance of the new approach is quantitatively evaluated with a specific focus on the affects of self occlusion on the alignment process. Experimental results are presented for simulated self-occlusions, and for the Bosphorus and UMB datasets, which both include occluded and self-occluded faces. The chapter concludes with discussion and conclusions in section 6.5.

6.2 Nasal signature operators

The nasal region has some distinctive features which make it one of the most important parts of the face for both recognition and detection. Its saliency and convexity make it easily detectable and it also has relative stability and rigidity over various expressions [67]. Before explaining the alignment procedure the two operators which are used in the pose correction algorithm are explained in detail.



(a)

(b)

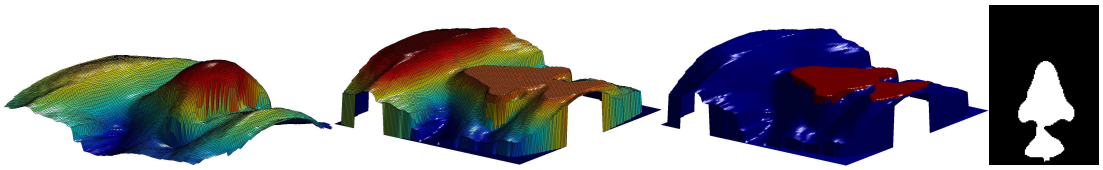
Figure 6-1: (a) A set of unaligned and distorted images; From left to right: a simple roll rotation, occlusion, unsymmetrical and symmetrical missing data. (b) The iterative PCA alignment results of (a). Except for the first and last columns, the PCA alignment fails as the distortion affects the orientation of the principal axes.

6.2.1 Inverted filling operator

Similar to chapters 4 and 5, the $\mathbf{IFill}(.)$ operator is applied to the nose region. The result is the binary image $\mathbf{B}$, which is the same size as the input depth map $\mathbf{Z_n}$, of the nasal region. This procedure is illustrated in Fig. 6-2. The face is denoised and the nose region cropped using the procedure described in [2], giving the input image shown in Fig. 6-2-a. The result of morphological filling, applied to $-\mathbf{Z_n}$, and the largest connected region in the filled image are shown in Fig. 6-2-b and -c, respectively. Extracting the connected region, in which the nose tip is located, gives the binary image $\mathbf{B}$, that is plotted in Fig. 6-2-d.

6.2.2 Parallel plane intersection

The other operator used in the alignment algorithm is the parallel plane intersection operator. A plane parallel to the xy plane, whose depth is less than that of the nose tip is intersected with the nasal region. Labelling any closed contour, in which the nose tip is located will result in a



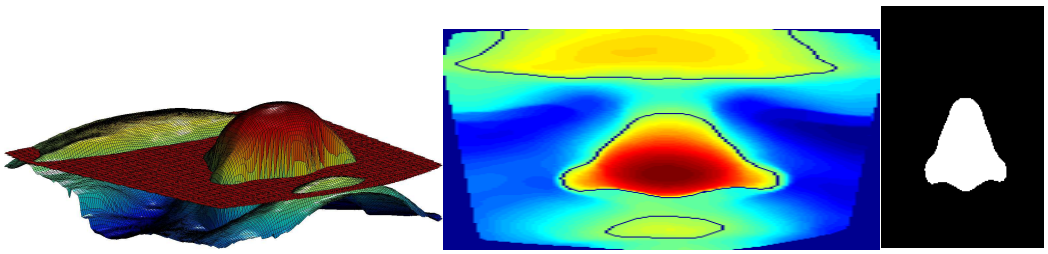
(a) (b) (c) (d)

Figure 6-2: The $\mathbf{IFill}(.)$ procedure: (a) The inverted nose region ($-\mathbf{Z_n}$). (b) The morphological filling result. (c) The largest connected region of the filled image. (d) The binary image $\mathbf{B}$ which is extracted from the labeled filled image (c).

binary image ($\mathbf{B^{int}}$),

$$\mathbf{B^{int}} = \mathbf{Int_D}(\mathbf{X_n}, \mathbf{Y_n}, \mathbf{Z_n}) \tag{6.1}$$

where $\mathbf{Int_D}$ is the operator which intersects the nose surface with the plane, having a $[0, 0, 1]$ normal and including the point $\mathbf{T} - [0, 0, D]$ ($\mathbf{T}$ denotes the nose tip). This operation is plotted in Fig. 6-3. The parallel plane is intersected with the nasal surface, resulting in two closed and one open contours (Fig. 6-3-b). Detecting the contour, which includes the nose tip and labelling its inner part creates the binary image $\mathbf{B^{int}}$, shown in Fig. 6-3-c.



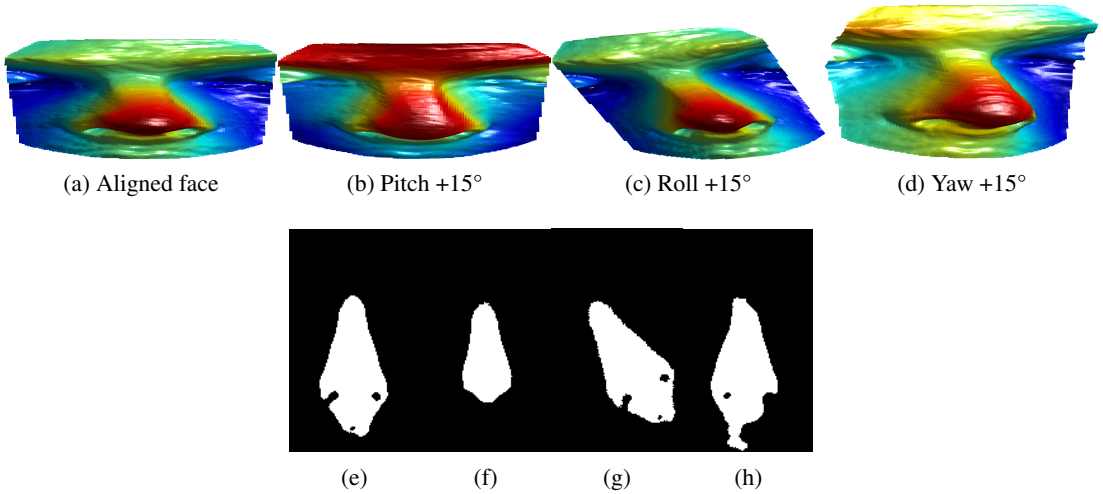
(a) (b) (c)

Figure 6-3: (a) Parallel plane intersection applied over the nasal region; (b) The intersection results as contours; (c) $\mathbf{B^{int}}$, which shows the inner parts of the contour in which the nose tip is located.

6.3 The self-alignment procedure

The starting point for the proposed self-alignment technique is the relationship between the face alignment and the binary image $\mathbf{B}$, in particular the observation that the face can be consistently aligned such that both the area and vertical symmetry of $\mathbf{B}$ are maximized. This is illustrated

in Fig. 6-4 where the aligned face and its binary image $\mathbf{B}$ are shown in Fig. 6-4-a and -e, respectively.



(a) Aligned face (b) Pitch +15° (c) Roll +15° (d) Yaw +15°

(e) (f) (g) (h)

Figure 6-4: The effect of rotations on the binary image $\mathbf{B}$. (a) Aligned nose region; (b), (c) and (d) rotations around the x, y, and z axes, respectively and (e)-(h) the corresponding binary images.

A change in pitch reduces the area of $\mathbf{B}$, as shown in Fig. 6-4-b and -f, while variations in alignment caused by rotations in the roll and yaw directions (Fig. 6-4-c and -d) result in reduced vertical symmetry of $\mathbf{B}$ (Fig. 6-4-g and -h). The aligned face of Fig. 6-4-a has the largest and most symmetrical binary image. This is the general case for all face images and is the premiss of the self-alignment technique. In the next step, an objective function that quantifies the area and symmetry of $\mathbf{B}$ is defined and then the three parameters, θ_x, θ_y and θ_z, the rotational angles around the x, y and z axes, respectively, are iteratively varied such that its global minimum gives the rotations around each axis required to align the face images. The definition of this function is discussed in the next section.

Although this procedure can successfully correct the pose variations around the yaw and roll directions, the pose variations around the pitch direction are more problematic. This is mainly due to the depth variations caused by hair on the forehead, mustaches and extreme facial expressions. This problem can be resolved by performing further tuning of the pitch rotation. Similar to the previous step, an objective function is introduced and minimised by rotating the nose region around the $\mathbf{x}$-axis. This will be explained in section 6.3.2.

The objective functions are defined in subsections 6.3.1 and 6.3.2. Then, in section 6.3.4, the method used to detect their global minima is explained.

6.3.1 Roll, yaw and coarse pitch poses

First, the **IFill** function is applied to the nose region depth map, resulting in the binary image $\mathbf{B}$. The line of symmetry which divides $\mathbf{B}$ into its two most similar halves is then detected. Detecting this line accurately is crucial for the alignment procedure and procedure used to perform this task. The sum of all rows of $\mathbf{B}$ is calculated and the column with the maximum value is considered to be the symmetry column. This procedure is shown in Fig. 6-5. One advantage of using this approach for symmetry column detection is that it can be localised even if, due to an imprecise cropping, $\mathbf{B}$ is translated along $\mathbf{x}$ or $\mathbf{y}$ directions. Therefore, the approach is invariant to 2D affine translation, unlike less robust methods such as using the nose tip location or finding half of the number of columns.

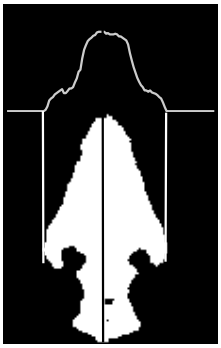


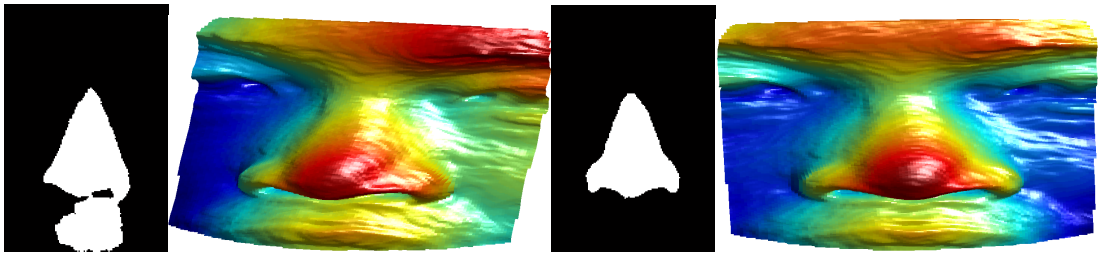
Figure 6-5: The two symmetry column detection methods illustration: The vertical white lines denote the two farthest columns which can include $\mathbf{B}$. The gray curve at the top is the sum of each column whose maximum location gives the solid vertical line.

The aim is to obtain the most symmetrical and largest $\mathbf{B}$ by rotating the nose in 3D. However, as mentioned earlier, some unwanted depth variations can negate the hypothesis that the largest area of $\mathbf{B}$, results when the pitch pose is correct. In order to fix this issue, the area of $\mathbf{B}$ is first limited by calculating the **AND** with $\mathbf{B^{int}}$ by,

$$\mathbf{B}' = \mathbf{AND}(\mathbf{B}, \mathbf{B^{int}}). \tag{6.2}$$

The left and right halves of $\mathbf{B}'$ ($\mathbf{B^l}$ and $\mathbf{B^r}$) are folded and their logical **AND** computed. It is expected an aligned face will have the maximum overlap of the left and right halves **AND**. A cost function whose optimum point corresponds to the best overlap can be defined as,

$$E_1 = -\left(\sum_{\mathbf{i}\prime,\mathbf{j}\prime} \mathbf{AND}(\mathbf{B^l}, \mathbf{B^r})\right), \tag{6.3}$$



(a) (b) (c) (d)

Figure 6-6: (a) **B** found by the E_1's global minimum and (b), its corresponding 3D nose region. (c) **B** found by the E's global minimum and (d), its corresponding 3D nose region.

in which $\mathbf{i}\prime = [1, \ldots, M\prime]^T$, $\mathbf{j}\prime = [1, \ldots, N\prime]^T$, and $M\prime$ and $N\prime$ are the number of rows and columns in the cropped nose image. When E_1 is at its global minimum, the two halves of $\mathbf{B}'$ have the highest overlap. However, there are some cases in which this hypothesis fails. The **AND** operation only considers the foreground of $\mathbf{B}'$ halves (the white pixels), by applying the sum over their overlapping region. However, for some noses $\mathbf{B}'$ is not necessarily fully symmetrical, but can still generate a high value of E_1. An example of this case is shown in Fig. 6-6. In Fig. 6-6-a, the result of the **IFill** algorithm for the global minimum of E_1 is shown. Although the **AND** of the left and right halves of this image are maximised, it does not result in an align face (Fig. 6-6-b).

The reason for it is that the symmetry criterion needs to consider both background and foreground pixels and this is achieved by the modified energy function,

$$E_2 = \left(\sum_{\mathbf{i}\prime,\mathbf{j}\prime} \mathbf{XOR}(\mathbf{B}^{\mathbf{l}}, \mathbf{B}^{\mathbf{r}}) \right), \tag{6.4}$$

where the **XOR** maximises the symmetry by finding the exclusive **OR** of the halves. The **XOR** operator finds the unequal parts of the binary halves and is maximised when both the background and foreground are most similar. Compared to the **AND** operation, the **XOR** is much appropriate for symmetry evaluation. The global minimum of E_2 corresponds to the most symmetrical $\mathbf{B}'$ that can be obtained by rotating the 3D nose region. However, minimising E_2 is not sufficient for successful alignment, as the **XOR** of the halves is independent of the regions' area. Even a very small but unsymmetrical region can have a high **XOR**. Therefore, combining the objective functions E_1 and E_2 will simultaneously maximise both the area and the symmetry of **B**.

The simplest approach to combining E_1 and E_2 is to find their sum,

$$E_{T_1} = \sum_{\mathbf{i\prime,j\prime}} \mathbf{XOR}(\mathbf{B^l}, \mathbf{B^r}) - \sum_{\mathbf{i\prime,j\prime}} \mathbf{AND}(\mathbf{B^l}, \mathbf{B^r}). \tag{6.5}$$

Although the addition works relatively well for many 3D faces, it fails for the cases in which either E_1 or E_2 is very large. These instances would occur when $\mathbf{B}'$ is small but highly symmetrical (low E_1 but high E_2). Another example is a very large but unsymmetrical foreground in $\mathbf{B}'$ (low E_2 but high E_1). In order to resolve this problem, a regularising factor can be inserted into the cost function equation. The factor should reduce the effects of E_1 or E_2 when they are unreasonably large. During the optimisation procedure, whenever the ratio $\frac{E_2}{|E_1|}$ is large, the influence of E_2 on the combination should be reduced and vice versa. The final energy function capable of handling these cases is as follows,

$$E = \alpha E_2 + (1 - \alpha) E_1 \tag{6.6}$$

where $\alpha = \frac{E_2}{|E_1|}$, and E_1 and E_2 are defined by (6.3) and (6.4), respectively. The value of α should be between zero and one, since the result of **XOR** is bigger than **AND**.

6.3.2 Fine tuning of pitch rotation

In order to tune the rotational variations around the x-axis, which can result from intense facial expressions, another objective function is introduced, which is used to make $\mathbf{B}$ and $\mathbf{B^{int}}$ as similar as possible. The desired pitch pose is defined as the one, which produces the most similar $\mathbf{B^{int}}$ and $\mathbf{B}$. Therefore, the following objective function is,

$$E_P = \sum_{\mathbf{i\prime,j\prime}} \mathbf{XOR}(\mathbf{B^{int}}, \mathbf{B}). \tag{6.7}$$

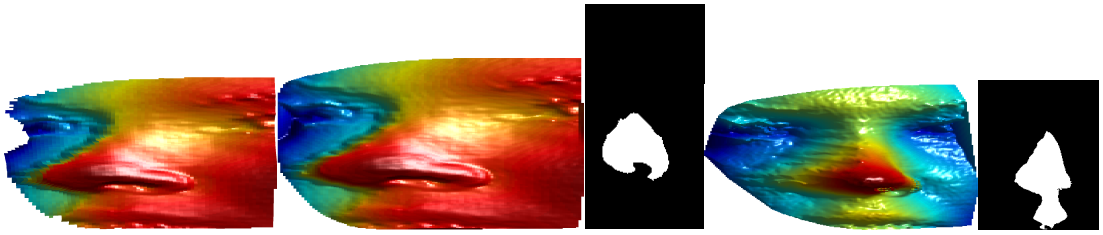
Similar to (6.4), the $\mathbf{XOR}$ operator is utilised to compare the dissimilarity of the binary images. By varying θ_x, the optimum point of the function is found which corresponds to the correct pitch alignment. The intersection image ($\mathbf{B^{int}}$) helps to bound $\mathbf{B}$'s area, by avoiding the unwanted depth variations caused by facial surface deformations.

6.3.3 Dealing with self-occlusion

When self-occlusion occurs due to either yaw or pitch rotations, some parts of the nose region are lost. In order to apply the alignment algorithm to self-occluded samples, as the nasal region is rotated during the alignment procedure, this missing data resulting from self-occlusion is replaced using Delaunay triangulation-based interpolation and the surface is resampled using

a uniform grid with 0.5 (mm/pixel). Finally, the **IFill** operator is similarly applied to the depth map $\mathbf{Z_n}$, by inverting it and performing the morphological filling.

Although self-occlusion distorts $\mathbf{B}$, the same criterion still stands, i.e. the most symmetrical and largest $\mathbf{B}'$ aligns the face. For example, the procedure for finding $\mathbf{B}$ for an image in the Bosphorus dataset, with 45°yaw rotation is shown in Fig. 6-7. The interpolation and resampling result is plotted in Fig. 6-7-b and the corresponding binary image $\mathbf{B}$ is shown in Fig. 6-7-c. The aligned nose and its corresponding binary image $\mathbf{B}$ are shown in Fig. 6-7-d and -e.



(a) (b) (c) (d) (e)

Figure 6-7: An example of dealing with self-occlusions: (a) The cropped 3D nose region which is self-occluded due to 45°yaw rotation. (b) Replacing the invalid points by the median of the valid points. (c) The binary image $\mathbf{B}$ for the self-occluded image. (d) The aligned image and (e) its binary image $\mathbf{B}$.

6.3.4 Optimisation using pyramidal SA

The optimisation problem is to find the global minimum of E in (6.6) in a three dimensional parameter space $[\theta_x, \theta_y, \theta_z]$ for the first alignment and the one dimensional space for tuning the pitch (θ_x). The objective functions have many local minima, making it nearly impossible to use traditional gradient based algorithms to find the optimum point. Therefore, the well-known global optimisation approach SA is used instead. The initial temperature (T_0) is chosen to be sufficiently high (in this work, 10000) and is very slowly decreased in each iteration. When the difference of the current point (X_{i+1}) of the function is higher than the previous point (X_i), i.e. $\Delta E = (E_{i+1} - E_i) > 0$, a modified Boltzmann function is employed to decide whether to keep X_{i+1} or skip using [154],

$$P_k(\Delta E) = \frac{1}{1 + \exp(\frac{\Delta E}{T_k})} \tag{6.8}$$

where $P_k(\Delta E)$ is between 0 and 0.5, k is the iteration count and T_k is the temperature in the k^{th} iteration. When $\Delta E > 0$, a random value is uniformly chosen in this boundary and

compared with $P_k(\Delta E)$. If the random value was smaller, the previous point is replaced by the current point. This procedure is utilised in order to avoid the algorithm becoming trapped in local minima. Moreover, the point with the smallest value found is compared in each iteration with the current value and is updated if the current value is smaller.

In order to improve the optimisation algorithm's robustness and accuracy, a pyramidal algorithm is used. SA is run many times and over different boundaries, and the search boundary is decreased after each run. Each successive SA is initialised with the previously found point and fewer iterations are used. Therefore, the algorithm will have the capability to more accurately tune the global minimum. This pyramidal approach can handle both large and small rotations. This procedure is depicted in Fig. 6-8. The temperature function is also shrunk in each layer of the pyramid, based on,

$$T_k = \frac{T_0}{2^L \log(k)} \tag{6.9}$$

where L is the number of the current layer. A lower temperature will result in a smaller value of $P_k(\Delta E)$ in (6.8) and this will decrease the probability that the lowest minimum found so far is updated.

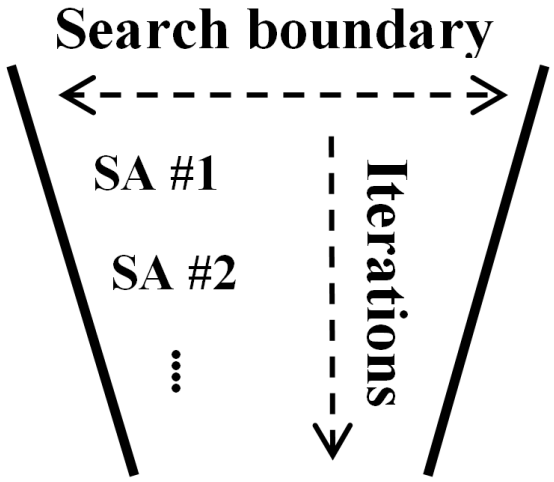


Figure 6-8: The pyramidal SA procedure for pose alignment (6.10); In each level of the pyramid (SA 1, 2, . . .) the search boundary and initial temperature are reduced and the initialisation is performed using the previous optimum point.

Therefore, when the cooling procedure is accelerated in each pyramid's level according to (6.9), the global minimum is tuned more properly. In subsequent experiments, SA is applied 6 times ($L = 0, 1, 2, \ldots, 5$). The search boundary in each iteration is chosen around the previous optimum point using the following equation (for $L > 0$),

$$\mathbf{S}_L = \frac{\mathbf{S}_{L-1}}{2^L}. \tag{6.10}$$

where $\mathbf{S}_L$ is the search boundary in the L^{th} level and $\mathbf{S}_0 = [-\pi/3, \pi/3]$. Therefore, $\mathbf{S}_5 = [-\pi/96, \pi/96] \approx [-1.9, 1.9]$ (degrees), which enables the algorithm to finely adjust the global minimum with approaximately 2.82°precision. Since $\mathbf{S}_0 + \mathbf{S}_1 = [-\pi/2, \pi/2]$, the algorithm is potentially capable to detect the coarse angular variations along the pitch, yaw and roll directions in the first two pyramid's levels. A block diagram of the whole procedure is depicted in a block diagram in Fig. 6-9. Calculating the **AND** of $\mathbf{B}^{\mathbf{int}}$ and $\mathbf{B}$ in (6.2) significantly helps avoid incorrect rotations. For instance, if in each iteration, the current rotation angles move the face in a completely wrong direction, no closed contours will be detected and, therefore, $\mathbf{B}'$ will be an empty image. As a consequence, E_1 will be small, but E_2 will be large. It will result in a large α, a large E, and the current parameter is rejected in the SA search space. Finally, the overall algorithm is depicted in Fig. 6-10 in a block diagram.

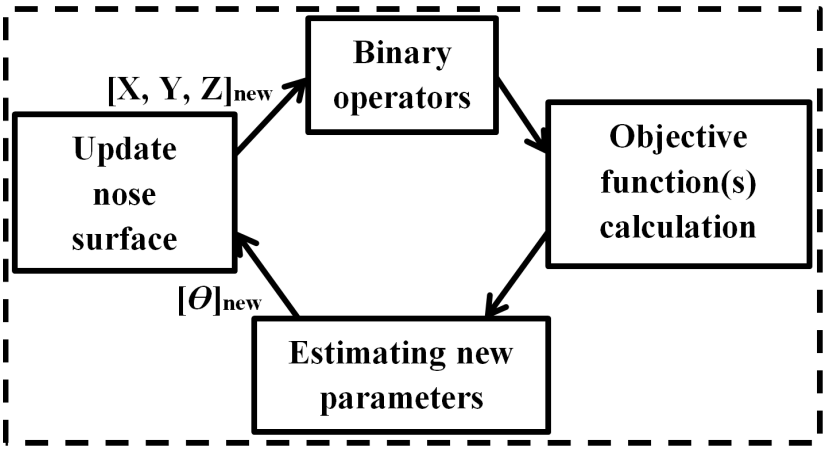


Figure 6-9: The block diagram of the alignment algorithm $\mathbf{Z_{new}}$ is the updated depth map using the angles found.

6.4 Experimental results

The alignment algorithm has been evaluated on three 3D face datasets, which are the FRGC v2.0, UMB dataset and the Bosphorus datasets. The faces in the Bosphorus dataset have large pose and expression variations. The rotations are 10°, 20°, 30°, 45°and 90°around the yaw direction, which result in self-occlusion. These angles are considered as the ground truth, which makes the dataset quite appropriate for the quantitative evaluation of the alignment algorithm.

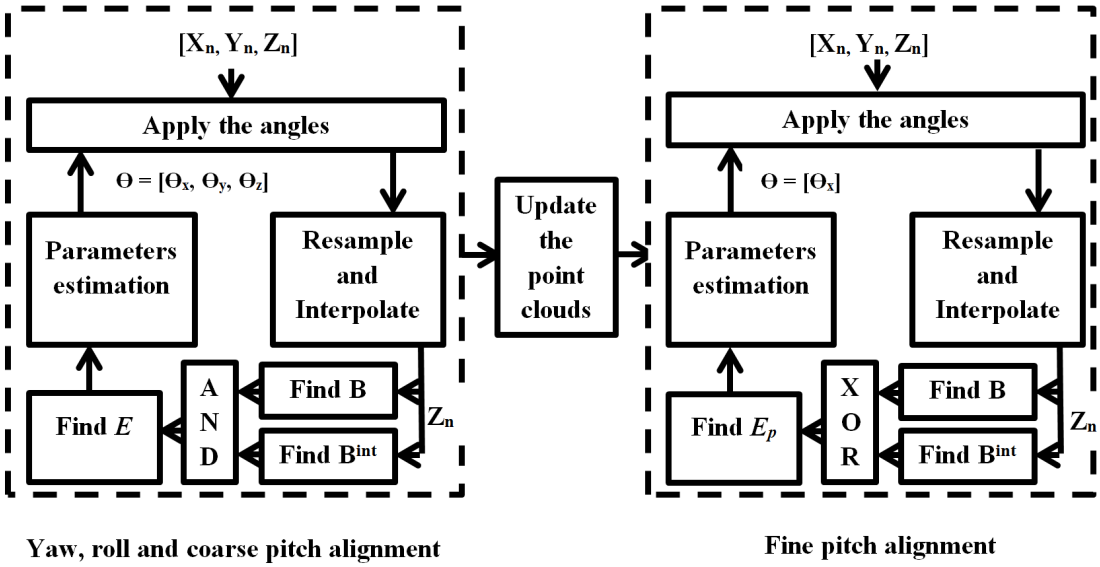
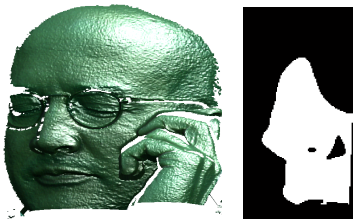


Figure 6-10: The overall block diagram of the proposed approach, which first minimise E in (6.6) and then E_P in (6.7), to correct the pose around the yaw and roll, and pitch, respectively.

In section 6.4.1, some examples of rotated faces and their alignment procedure are provided. In the next sections, these datasets are used for the following purposes: in section 6.4.2, the FRGC dataset is utilised for within-class consistency analysis; the UMB dataset is used in section 6.4.3 for evaluating the algorithm's robustness against occlusion; finally the algorithm is applied to the Bosphorus dataset in section 6.4.4 to quantitatively assess the proposed pose correction approach for self-occlusions. The Bosphorus dataset is used in section 6.4.5 to examine the scale invariant feature of the algorithm and its advantages for computational speed.



(a) (b)

Figure 6-11: Initial pose showing that the symmetry of the binary image has been degraded due to the pose variation and occlusions: (a) the input 3D face; (b) **B** in the first iteration.

6.4.1 Some pose correction examples

Figure 6-11-a shows an example input image selected from the UMB dataset with occlusions due to the hand and glasses and self-occlusion caused by yaw rotation. In the initial state of $\mathbf{B}$ depicted in Fig. 6-11-b the asymmetry is mainly due to the yaw rotation. E is iteratively calculated over $\mathbf{B}$ according to (6.6) and its optimum parameters are found by the pyramidal SA algorithm.

Figure 6-12: $\mathbf{B}$ at E's global minimum found after (a) 10^{th}, (b) 20^{th}, (c) 50^{th}, (d) 150^{th} and (e) final iterations; (f) shows the aligned face.

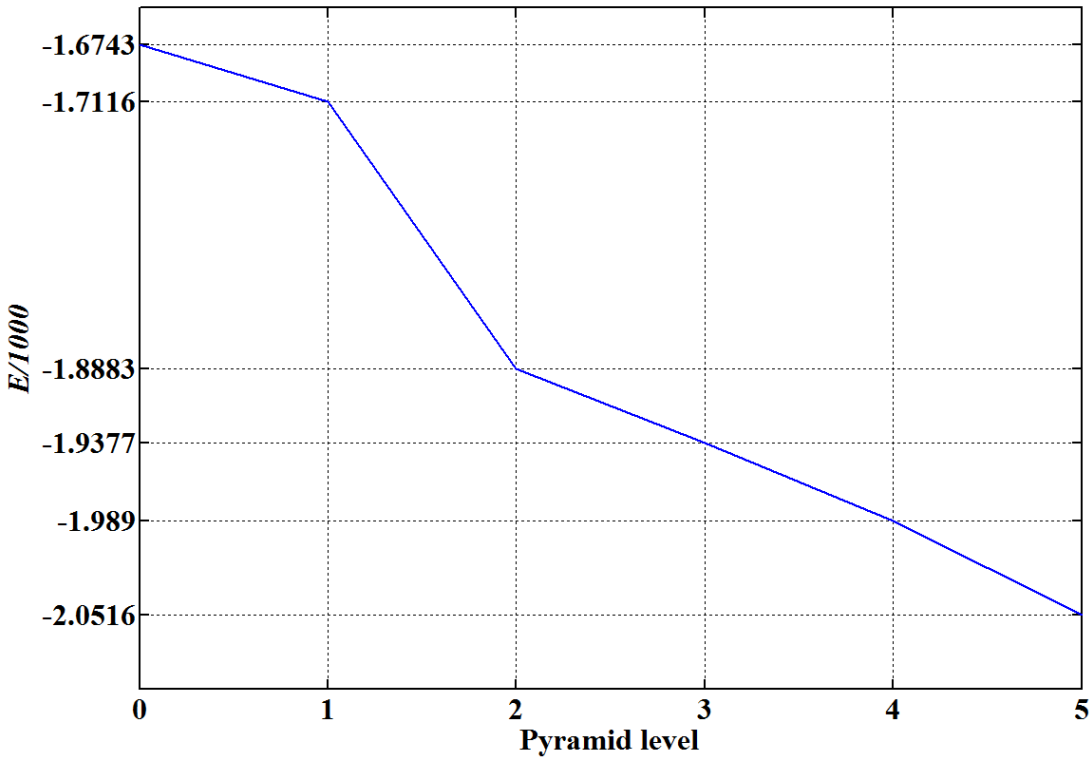


Figure 6-13: The optimum values of E in different levels of the pyramid.

The binary image $\mathbf{B}$ for the best candidates found after 10, 20, 50 and 150 iterations are shown in Fig. 6-12-a to -d. The final binary map and alignment result are shown in Fig. 6-12-b and -c, respectively. Convergence is reached after 591 SA iterations and the final value of E is -2051.6. As it can be seen in Fig. 6-12-e, the largest and most symmetrical image for $\mathbf{B}$ is found in the last iteration. In this example, the optimum values found for the angles were $\theta_x = +1.2254°$, $\theta_y = +24.8545°$and $\theta_z = +0.2632°$. As was expected, θ_y has the highest value, due to the significant pose variation around the yaw direction.

As mentioned in section 6.3.4, the optimisation is performed in six pyramidal levels. The best point found by the previous optimisation is assigned as the starting point for the current SA. The range of the search boundary is also reduced in each SA run. The whole optimisation procedure takes 591 iterations, with 300, 150, 75, 37, 19, 10 for the zeroth to fifth level, respectively. The values of E at its global minima in different pyramid's levels are shown in Fig. 6-13.

Figures 6-14 and 6-15 show other examples of the consistency of the alignment algorithm over different expressions. The samples are selected from the UMB dataset. The alignment results and the binary image $\mathbf{B}$ at the global minimum are shown.

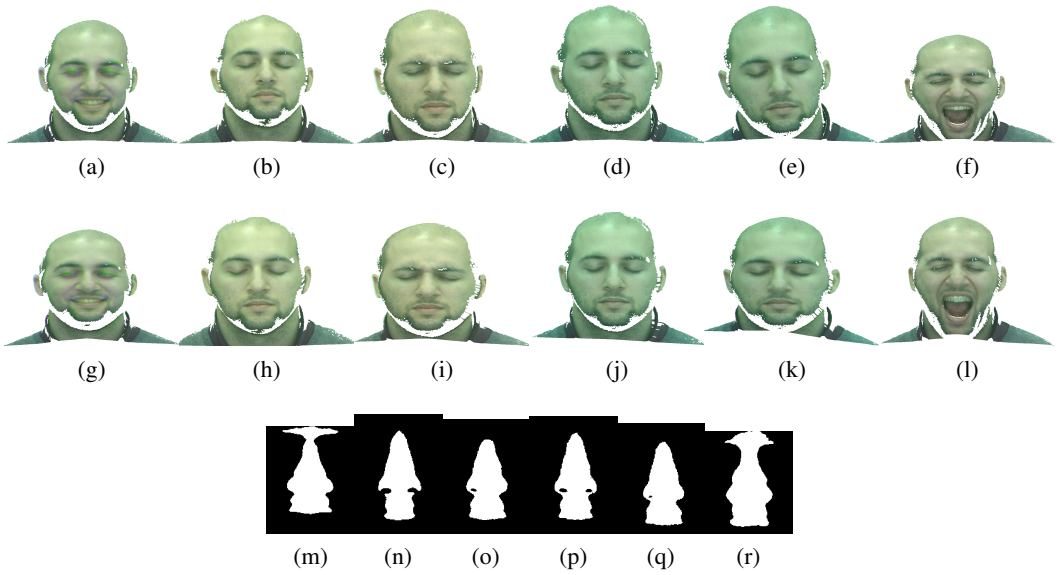


Figure 6-14: Example of alignment for different expression and poses: (a-f) The input unaligned faces. (g-l) The alignment result. (m-r) $\mathbf{B}$ at the optimum point.

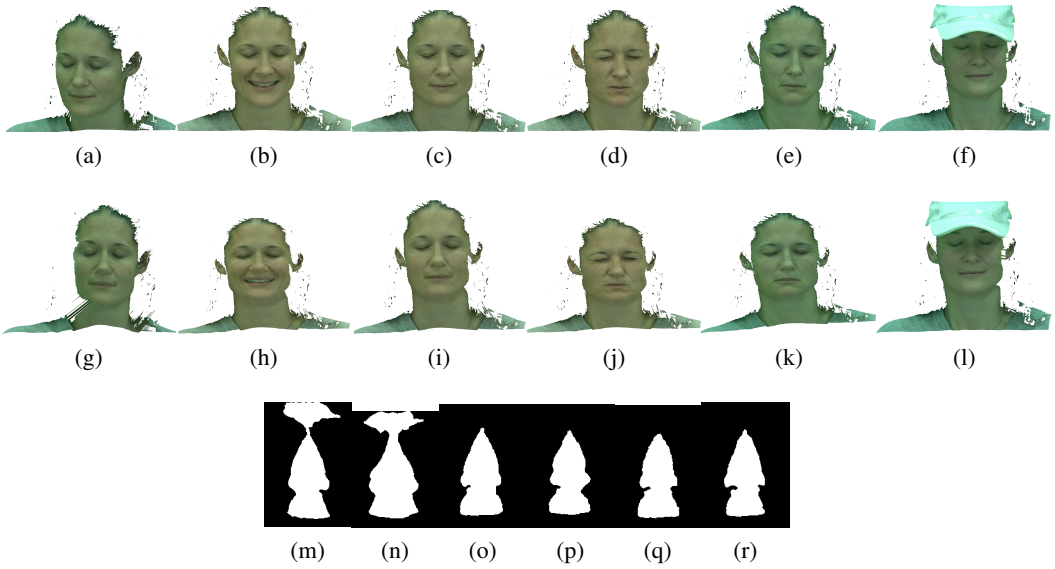


Figure 6-15: Example of alignment for different expression, poses and occlusion: (a-f) The input unaligned faces. (g-l) The alignment result. (m-r) $\mathbf{B}$ at the optimum point.

6.4.2 Within-class consistency

As discussed above, the alignment algorithm mainly relies on the area and symmetry of the binary image $\mathbf{B}$ computed from the nose region. Although most noses are not exactly symmetrical, it is still the case that the most symmetrical and largest $\mathbf{B}$ can be found. Since the nose structure remains largely constant for each subject, the alignment always generates similar results. This is what is expected from a facial pose correction algorithm that maintains the intra-class similarity. That is, the images in the same class should have similar pose after alignment. Therefore, even if the nose is not symmetrical for a specific subject (and as a consequence, $\mathbf{B}$ is not symmetrical), the optimum pose found from the global minimum of E should be very similar for all the samples of the subject.

In order to evaluate the consistency, an algorithm similar to the previous chapter is used using the ICP algorithm [1]. First, the faces in the FRGC dataset are all aligned using the proposed algorithm. Then, for each subject, one of the images is randomly selected as the reference and all of the other images are aligned to it using the ICP algorithm. Then, the Euclidean distance, E_r, between the rotation matrix and the identity matrix is computed by,

$$E_r = ||\mathbf{R} - \mathbf{I}||_2 \tag{6.11}$$

where $\mathbf{R}$ is the rotation matrix calculated by the ICP algorithm. Ideally, with perfect aligned images $E_r = 0$; However, in practice due to the noise, expression variation or alignment errors,

E_r will be greater than zero. The mean of E_r and $\mathbf{R}$ are calculated for all subject's samples and this procedure is repeated for all of the 557 subjects in FRGC v2.0. The results for E_r and $\mathbf{R}$ are plotted in Fig. 6-16 and 6-17, respectively.

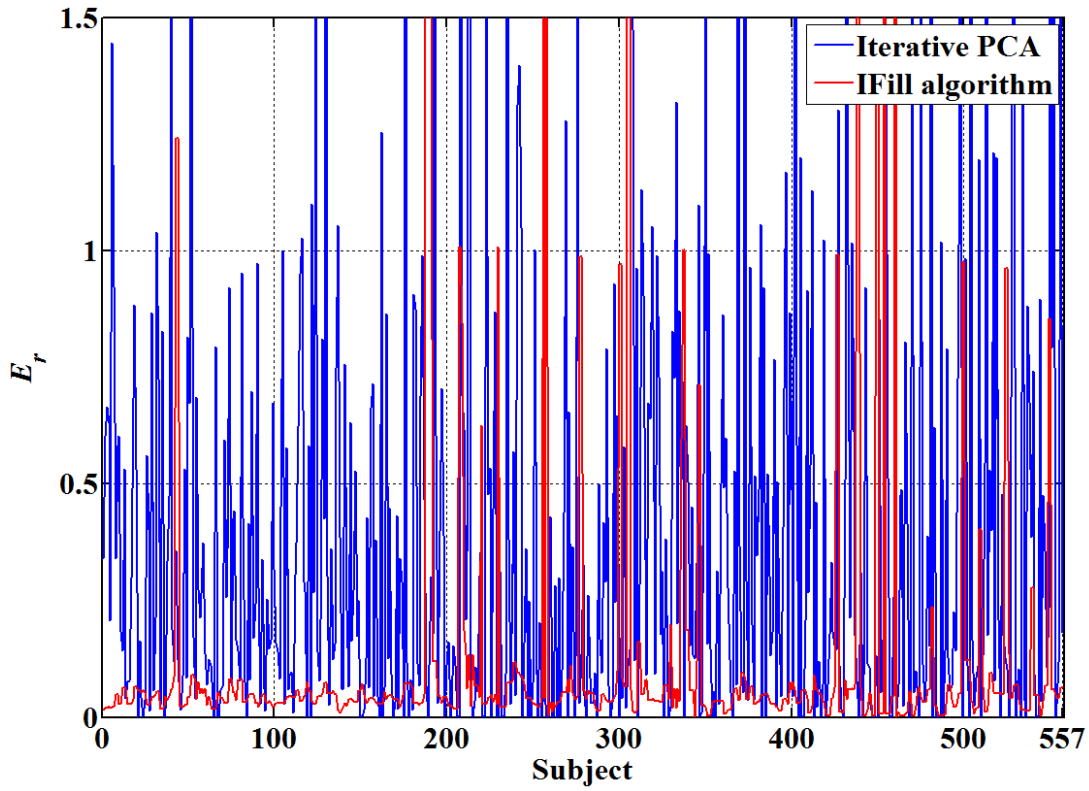


Figure 6-16: E_r calculated for all of 557 subjects in FRGC v2.0 using the proposed method and [32] alignment algorithm.

$$\begin{bmatrix} 0.9987 \pm 0.3367 & 0 \pm 0.1047 & 0 \pm 0.1117 \\ 0 \pm 0.1222 & 0.9992 \pm 0.1548 & 0 \pm 0.0593 \\ 0 \pm 0.1104 & 0 \pm 0.0431 & 0.9983 \pm 0.4008 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 0.9667 \pm 0.3822 & 0 \pm 0.12974 & 0 \pm 0.1845 \\ 0 \pm 0.1297 & 0.9933 \pm 0.1789 & 0 \pm 0.1066 \\ 0 \pm 0.1817 & 0 \pm 0.1194 & 0.9480 \pm 0.4402 \end{bmatrix}$$

(b)

Figure 6-17: Average ICP rotational matrix ($\bar{\mathbf{R}}$) for 557 subjects in FRGC dataset $\pm$ 1 std dev: (a) The proposed method; (b) Iterative PCA alignment [32].

Figure 6-16, verifies that the proposed approach is more consistent than the iterative PCA

algorithm, as the errors are significantly lower than for the PCA algorithm. The average of E_r for the new approach is 0.1283 which is approximately 0.33 times smaller than the average error for the PCA alignment average error, which is ≈ 2.98. Moreover, based on the result shown in Fig. 6-17, the average rotation matrix ($\bar{\mathbf{R}}$) for the aligned faces by the proposed approach is closer to the identity matrix than that of the PCA approach. Figure 6-17 also shows that the standard deviations for the values in $\bar{\mathbf{R}}$ are also lower for the proposed method.

6.4.3 Pose variation and occlusion

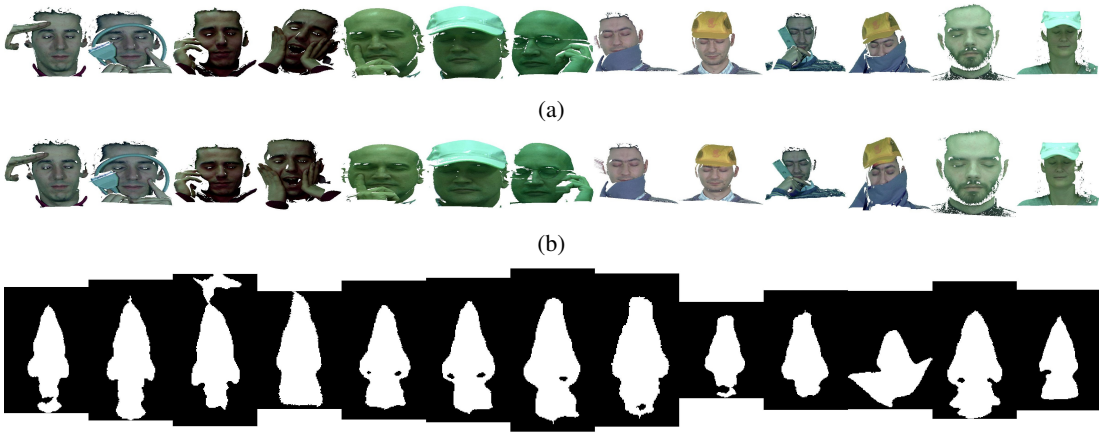
The proposed alignment algorithm is robust against partial occlusions of the face region. The usual causes of occlusions in biometric sessions are hair, hats, hands, glasses, and scarves (samples with beards and mustaches are also considered in this section). Occlusions can impulsively deform the depth uniformity. As the symmetry of data can be lost, PCA-based alignment algorithms fail, because the principal axis can not be correctly found. Moreover, the occluded parts can be assumed to be outliers in the data, which the ICP algorithm is highly sensitive to.

In this section, the new algorithm is applied to the UMB dataset's occluded and rotated samples. The dataset includes many occluded faces making it suitable for the approach robustness evaluation. Example alignment results, including the pose corrected faces and $\mathbf{B}$ at E's global minimum, are shown in Fig. 6-18 for a set of selected samples showing occlusions with different objects with (extreme) expression variations and rotations around the x, y and z axes[1]. The rotations around the yaw and pitch directions result in self-occlusions, which is why missing data is exposed after pose correction.Providing the nose tip is detectable and not occluded, and the nose region is not occluded (or relatively symmetrical occluded - some examples of this case are depicted in Fig. 6-18), the proposed method can successfully align the faces. However, if the occlusion is unsymmetrical over the nose region, the occluded parts are cropped as within the nasal region and this will degrade the symmetry of $\mathbf{B}$. In other cases, where different parts of the face are occluded (as shown in Fig. 6-18), the algorithm can correct the pose variations.

6.4.4 Robustness over rotation ranges

Robustness over a wide range of rotational angles is one of the most important factors of an alignment algorithm. In order to evaluate the detectable range of angles, objects are usually manually rotated to known angles and alignment is used to compute the amount of rotation. Although this is a standard approach for robustness evaluation for general 3D alignment methods,

[1]The algorithm has been tested on all images in the UMB dataset; However, other samples that are labeled as "not permissible to be published" by the dataset and the algorithm is capable of successfully align are not shown.



(a)

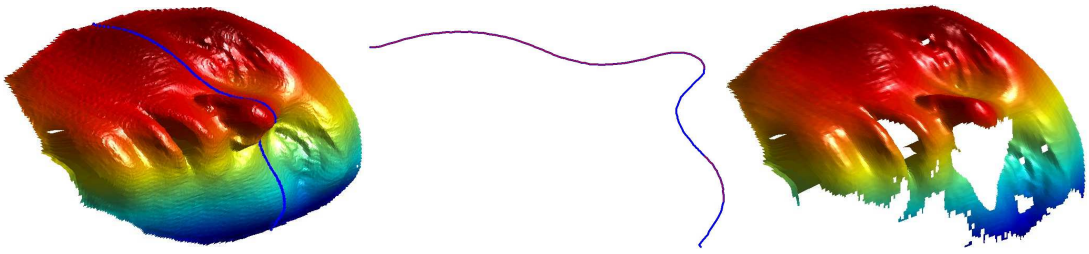
(b)

(c)

Figure 6-18: Occluded samples from the UMB dataset: (a) The input unaligned faces. (b) The alignment result. (c) $\mathbf{B}$ at the optimum point.

there is a specific difference in face acquisition in biometrics sessions. As the cameras locations are fixed, the rotation, which is usually around the yaw and pitch directions, can result in some parts of the face being occluded by the face itself. This problem is known as self-occlusion and to quantitatively evaluate the proposed method's robustness to self-occlusion, two approaches are used. The first one is based on artificially simulating self-occlusions. The second uses the Bosphorus dataset's rotated images including various yaw rotations. As the ground truth for both methods is available, the alignment error can be easily calculated. The artificial self-occlusion procedure is as follows. First, the face is manually rotated. The rotation is performed around the yaw and pitch directions (the roll rotations will not cause self-occlusion). Then, the 3D face is horizontally (row-wised) and vertically (column-wised) scanned. During the scanning process, planes perpendicular to the xz and yz planes are intersected with the face surface, which results in a set of 2D curves. For each curve, the x or y indexes that correspond to more than one value of z are detected and are labeled as invalid data.

Figure 6-19 shows this replacement procedure. A 3D face image which is manually rotated 45°along the yaw direction is shown in Fig. 6-19-a. An intersection of a plane perpendicular to the yz plane with the face surface is shown in blue. The parts of the intersection which are occluded by the face regions are plotted in blue in Fig. 6-19-b and are labeled as invalid data. All parallel intersections are similarly found and the labelling is performed. Finally, the resulting 3D face is resampled using a uniform grid, with $1mm$ resolution, to create the new $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ maps. Any points missed due to the resampling are replaced using 2D cubic interpolation (not the self-occluded parts, just the points situated inside the $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ maps). The self-occluded image is finally shown in Fig. 6-19-c.



(a) (b) (c)

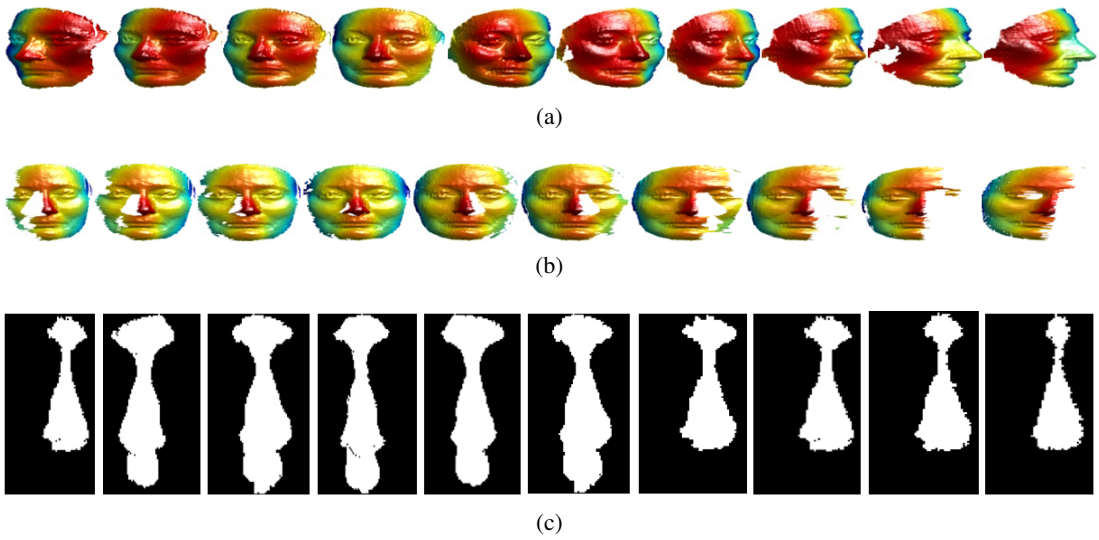
Figure 6-19: Artificial self-occlusion generation: (a) The intersection of a plane perpendicular to the **yz** plane. (b) Blue: the intersection; Red: after the repetitive parts are removed. (c) The result of self-occlusion.

The same routine is utilised for generating artificial occlusions around the pitch direction, the only difference being that for pitch self-occlusion, a vertical intersection is found. The alignment algorithm is then performed on the self-occluded face. After alignment, the argument error, err, is calculated by

$$err = |\theta_{\mathbf{g}} - \theta| \tag{6.12}$$

where θ_g is the grand truth pose and θ is the angles found by the proposed approach. For example, Fig. 6-20 shows the result of aligning a self-occluded face for different yaw angles. The rotated images are plotted in Fig 6-20-a. Artificial self-occlusion is applied and they are aligned using the proposed algorithm, giving the result shown in Fig. 6-20-b. The binary map ($\mathbf{B}$) at the objective function's global minimum is also shown Fig. 6-20-c. Even for the cases where data for nearly half of the face is missing due to the self-occlusion (such as 70°in Fig. 6-20), minimising E in (6.6) helps to capture the largest symmetrical area for $\mathbf{B}$. However, since most parts of the face and nasal region are lost for $|\theta_g|$ >75°, finding the biggest connected region in the filled image becomes impossible. This issue causes $\mathbf{B}$ to shrink and consequently, the area and symmetry hypotheses can not be satisfied.

In order to find the detectable angular range by the alignment algorithm, for each subject, the sample that looks exactly forward (no rotation) is selected. The above procedure is then performed to artificially generate self-occlusions. The angle θ_g is varied over the range [-90°,+90°] and the error is computed using (6.12). Figure 6-21 shows the average error for all of the subjects in the Bosphorus dataset. Due to the noise and slight unsymmetrical nature of faces, the error is not exactly symmetrical around 0°. The error is relatively stable in the range of [-50°,+50°] and even when $\theta_g \approx \pm 70°$, in which significant parts of one half of the face is nearly lost due self-occlusion, the error is only slightly higher than 4°(an example of this case is plotted in Fig. 6-20). The average error in the range of [-50°,+50°] is $\approx$ 0.3974°.



(a)

(b)

(c)

Figure 6-20: Artificial self-occlusion generation: (a) Yaw rotated self-occluded images : -50°, -40°, -30°, -20°, +20°, +30°, +40°, +50°, +60°and +70°. (b) The aligned faces. (c) $\mathbf{B}$ at E's global minimum.

For comparison, the iterative PCA alignment explained in [32] is also evaluated on the self-occluded images. First, one forward-looking sample for each subject is selected. Then, the self-occlusion is simulated for rotations around the yaw direction and the eigenvectors of the point's covariance matrix are computed. The eigenvectors can be assumed as a rotation matrix, which can be applied over the point clouds to align them on their principal axes [32]. Therefore, if the Euler angles are extracted from the rotation (eigenvector) matrix, they can be used to compute the error using (6.12) (the method explained in [155] is used to extract the Euler angles from the 3D rotational matrix). The process is repeated until the 3×3 eigenvector matrix equals to the identity matrix.

The red curve in Fig. 6-21 shows the result of the PCA-based alignment algorithm [32]. PCA fails when too many points are lost here due to self-occlusion. In particular, when the data loss is asymmetric (caused by yaw direction self-occlusion), the PCA alignment performance can be drastically degraded. Although it can align the faces rotated in the range of θ_g =[-20°,+20°] with $err \approx 1.25°$, the error substantially increases for higher rotations ($|\theta_g| > 25°$).

The self-occlusion is also simulated for the rotations along the pitch direction. The alignment result for a sample selected from the Bosphorus dataset is shown in Fig. 6-22-a, rotated along the negative and positive directions. In this figure, the faces are rotated from -50°to +80°in 10°steps and the alignment results are shown in Fig. 6-22-b. For comparison, the binary image $\mathbf{B}$ produced by the proposed **IFill** algorithm at the global minimum of E is plotted in Fig. 6-22. Similar to Fig. 6-21, for each subject in the Bosphorus dataset, the sample that looks directly forward is rotated around the pitch direction and the self-occlusions simulated.

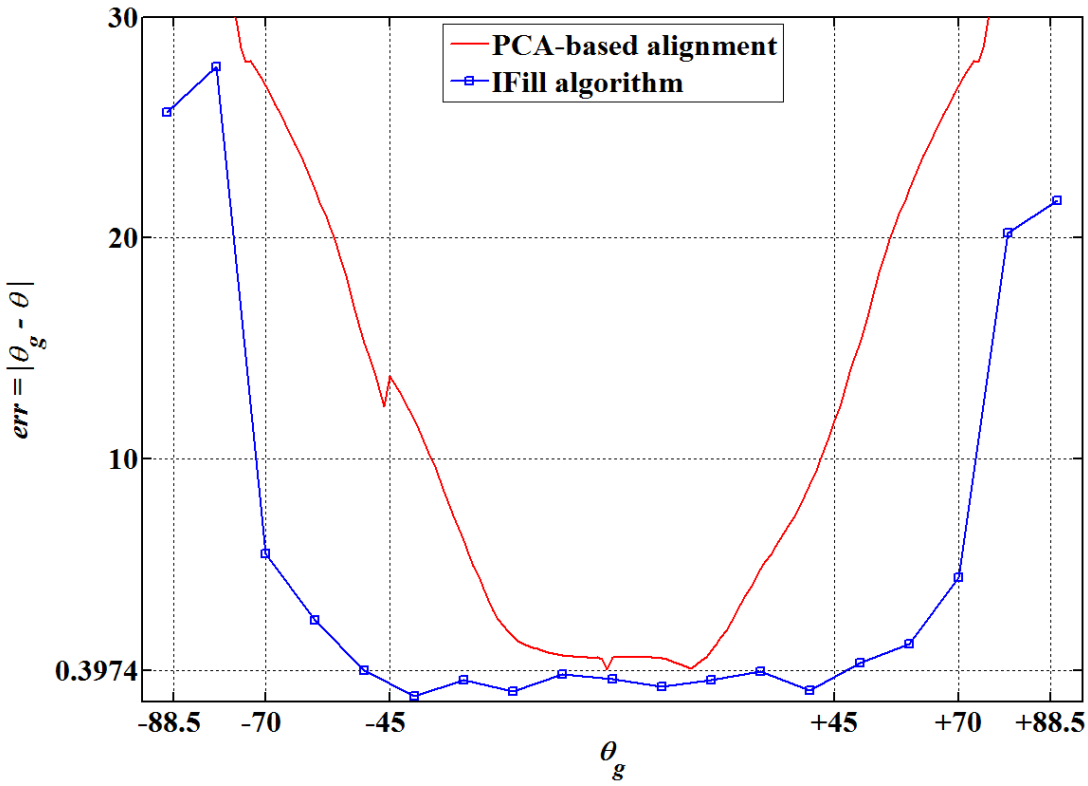
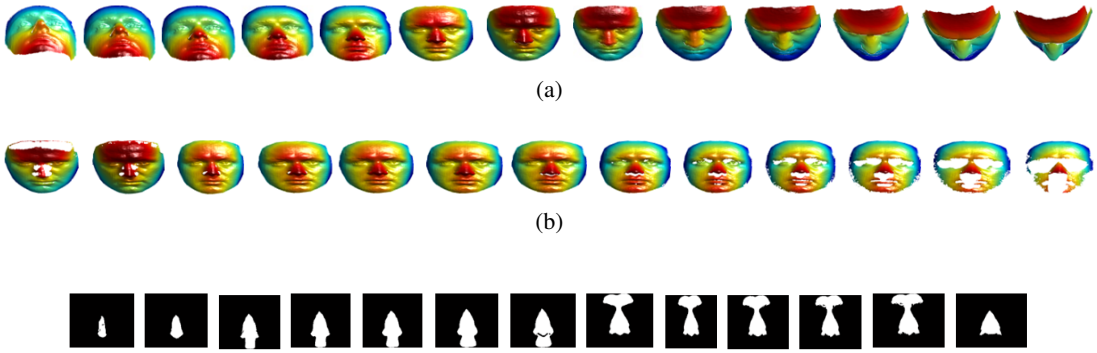


Figure 6-21: The alignment error for artificial self-occlusion along the yaw direction.

The angle θ_g is varied and the alignment error is calculated using (6.12). The result in Fig. 6-23 shows the detectable angular range is [-50°,+80°]. For $\theta_g < 50°$, the nasal region has significant occlusion, $\mathbf{B}$ is shrunk and the inverted filling **IFill** can not be calculated. An example of this case is in Fig. 6-22 for the leftmost first face. However, this will not however be a major problem in realistic 3D face acquisition sessions, since it is not feasible to rotate the face more than -50°around the pitch direction.

It should also be mentioned that the major alignment problem for $\theta_g < -50°$and $\theta_g > +80°$is the nose tip detection. In particular, for small (or flat) noses, the nose tip is very difficult to detect for these extreme rotations. The main reason is that for high values of $|\theta_g|$, the nose tip region is lost due to self-occlusion. Consequently, the chin (for $\theta_g < 0$) or forehead (for $\theta_g > 0$) are found as the biggest convex region after thresholding the mean and Gaussian curvatures. This problem does not occur for yaw rotations. Although the alignment algorithm successfully works for more salient and larger poses for even higher values of $|\theta_g|$, its detectable rotational range for the pitch direction is [-40°,+75°], with an error of $< 2.6°$.

The PCA-based alignment [32] is also plotted in red in Fig. 6-24. Compared to the data loss caused by yaw rotation (Fig. 6-21), PCA is more successful in pose correction for self-



(a)

(b)

(c)

Figure 6-22: Artificial self-occlusion generation: (a) Pitch rotated self-occluded images : -50°, -40°, -30°, -20°, -10°, +10°, +20°, +30°, +40°, +50°, +60°, +70°and +80°. (b) The aligned faces. (c) $\mathbf{B}$ at E's global minimum.

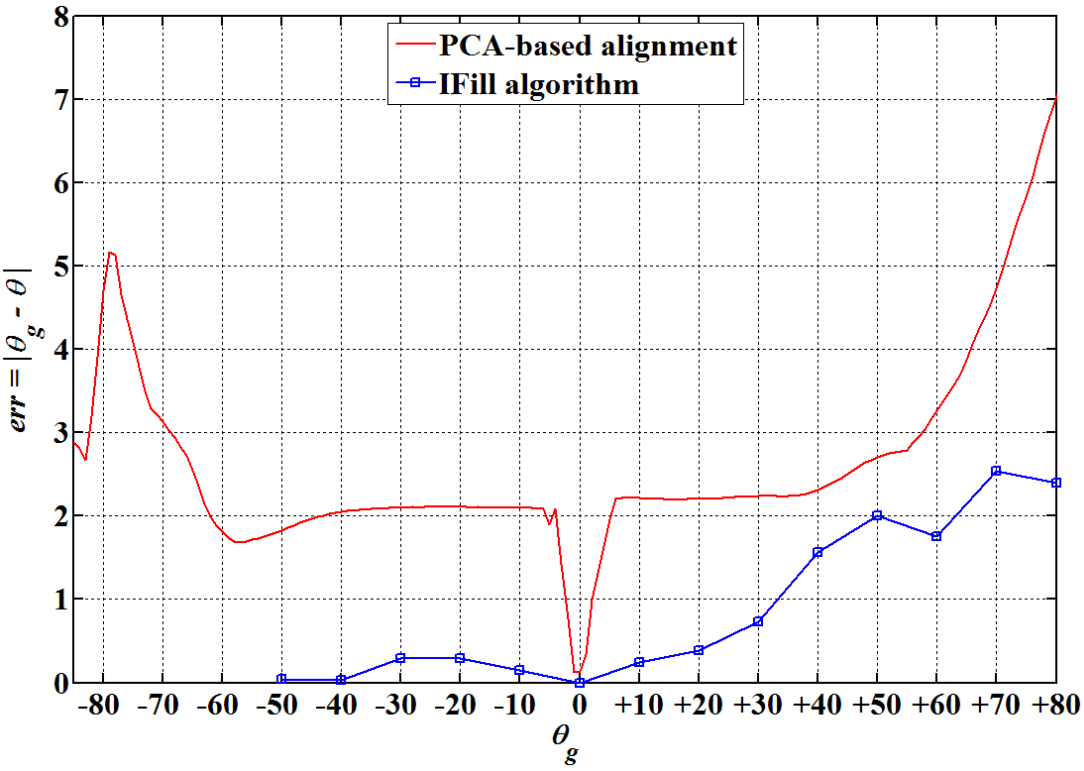


Figure 6-23: The alignment error for artificial self-occlusion along the pitch direction.

occlusions due to pitch rotation. This is because after the rotation around the pitch direction, the symmetry of the data distribution is still largely preserved and the principal axes can still

θ_g	**10°**	**20°**	**30°**	**45°**
IFill error	0.7792°	0.1719°	0.1915°	0.5352°
PCA-based [32] error	0.0774°	0.0110°	13.2558°	35.4615°

Table 6.1: The alignment error for the self-occluded faces in the Bosphorus dataset; Rotations are along the yaw direction.

be used to align the point clouds. This fact can be seen for the range θ_g = [-45°,+20°] in Fig. 6-23. However, when $\theta_g > 50°$(which means the subject is looking downward), since more data is occluded by the nose and cheek the error of the PCAalignment rapidly increases.

The other issue for discussion is the error at 0°, which would reasonably be expected to be zero. However, in both Fig. 6-21 and 6-23, it is > 0. This is because even for those samples in the dataset which are labeled as "not rotated", very small amounts of rotation exist. These minor rotations are detected by the proposed method and agglomerated for each subject in the average error, which finally results in non-zero errors for $\theta_g = 0°$.

The second approach used to evaluate the robustness of the proposed algorithm is to apply the alignment to the Bosphorus rotated images. As mentioned before, the dataset includes self-occluded faces. The subjects have intentionally rotated their faces in the yaw direction during the imaging sessions, 10°, 20°, 30°and 45°, during the imaging sessions. As the ground truth is again available, the same error function (6.12) can be calculated after alignment and is shown in Table 6.1. Again, the error increases with higher rotations, as would be expected. However, for 45°, the average error is just 0.5352°. Similarly, the PCA-based alignment is also applied to the rotated faces and its results are shown in the third row of the Table. 6.1. For smaller rotations, the PCA performance is more accurate. However, as the data loss increases, for larger rotations PCA fails.

The alignment on a selected sample in this category is shown in Fig. 6-24. The input images are shown in Fig. 6-24-a-d, the aligned images in Fig. 6-24-e-h and the binary images $\mathbf{B}$ for the optimum point are shown in Fig. 6-24-i-l. The example shows the success of the proposed alignment algorithm on natural self-occluded samples, as well as the artificial self-occluded ones.

6.4.5 Alignment in lower resolution

The proposed alignment algorithm is scale-invariant, which means it can be performed at any arbitrary resolution. Although the alignment only utilises the nose region, reducing the number of points via downsampling can still significantly speed up the algorithm. The downsampling, on the other hand, has the disadvantage of losing data. In this subsection the proposed algo-

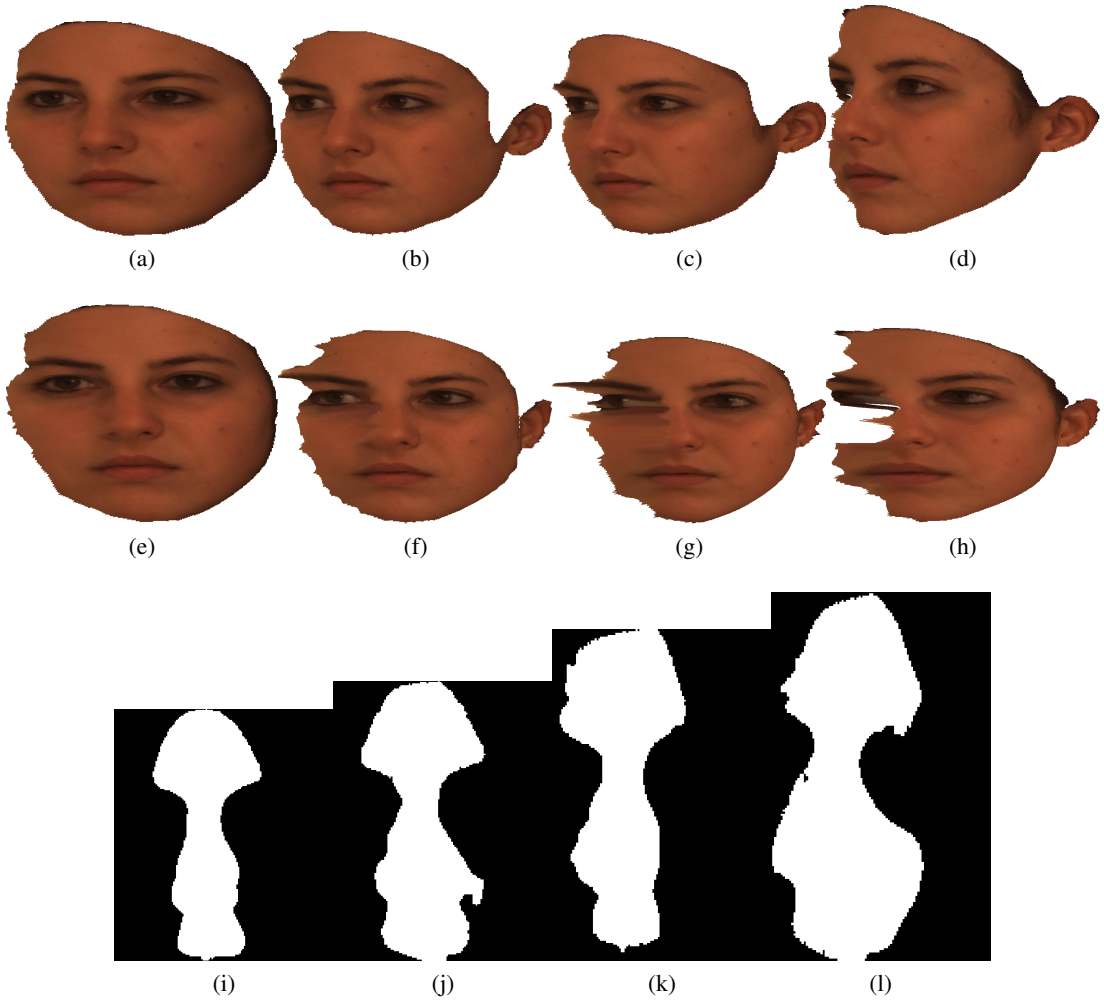


Figure 6-24: The alignment of self-occluded images by the proposed algorithm: (a-d) yaw rotated images for rotations of 10°, 20°, 30°, 45°. (e-h) The aligned faces. (i-l) $\mathbf{B}$ at E's global minimum.

rithm's capability to align 3D faces at lower resolutions and the benefits for its computational cost are evaluated. The minimum scale that can be used to acceptably to overcome the loss of data is also found.

For this purpose, the nose regions in those images in the Bosphorus dataset with $10°$, $20°$and $30°$rotations along the yaw direction are cropped. Then the resulting images are first resized to a fixed size (in this work 160×100). Then, the resolution is varied by downsampling, using Cubic interpolation and filtering to minimise high frequency aliasing. The alignment error is computed using (6.12) and the average error for all of the subjects is plotted in Fig. 6-25, for scale 0.1 to 1. The error remains relatively constant in the range of $[0.4, 1]$ for all

of the cases. However, for scales < 0.4, the error dramatically increases for the $30°$rotation, plotted in green. The error remains approximately stable around $2°$for smaller yaw rotations (the blue plot). Even for the case of $20°$rotation, the error for $\mathbf{Scale} = 0.3$ is less $< 2°$.

The data loss caused by self-occlusion becomes more significant in higher rotations as it inherently result in a small $\mathbf{B}$, as was shown in Fig. 6-20. Also, at lower scales the aliasing effects can more significantly degrade the downsampled image. These are the reasons for the drastic increase in error for $\mathbf{Scale} < 0.2$. However, the algorithm can correct the pose of the self-occluded images with an error of$<$ $1°$for scales in the range $[0.4, 1]$. Moreover, if the rotation of the faces in a given dataset is known a priori to be $< 10°$, then the alignment can be performed at even lower scales, with an error of $< 2.5°$. Finally, it is worth mentioning that for $\mathbf{Scale} = 1$, the error is $0.7792°$, $0.1719°$and $0.5352°$, for the case of $10°$, $20°$and $30°$yaw rotation, respectively, which are the same as Table 6.1.

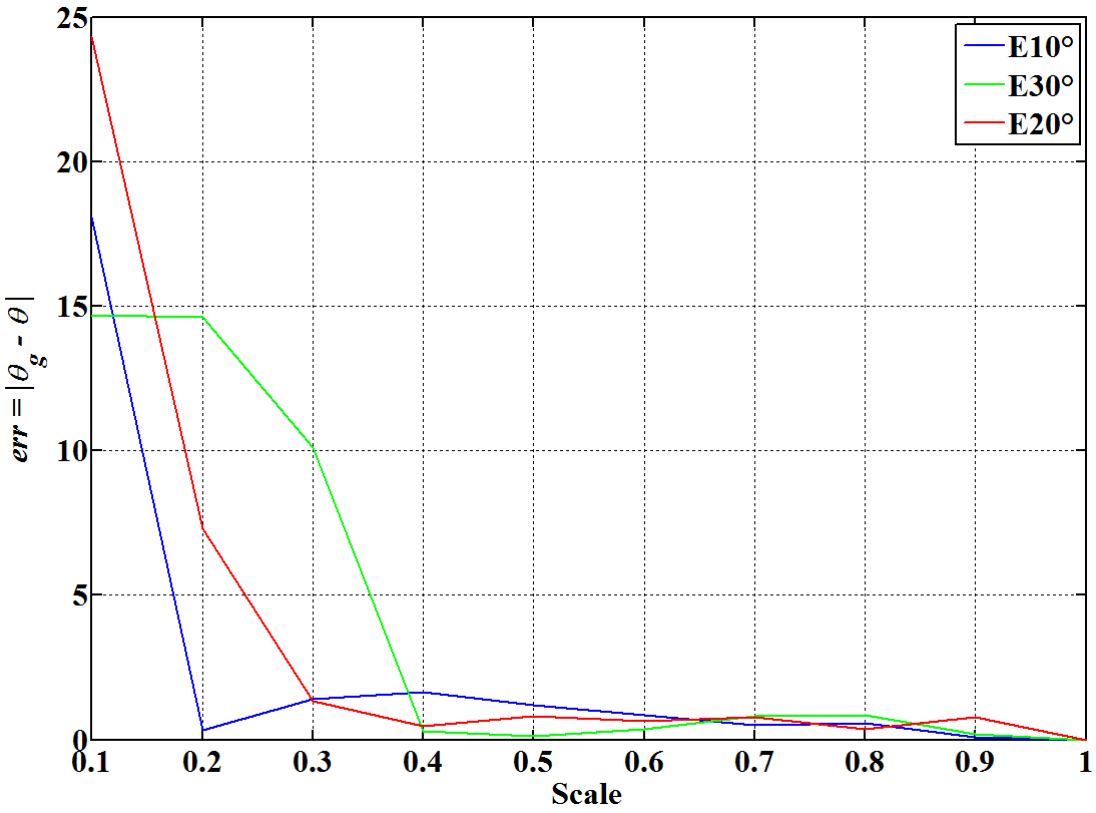


Figure 6-25: The average alignment error in different scales for $10°$, $20°$and $30°$self-occluded images in the Bosphorus dataset; $\mathbf{Scale} = 1$ corresponds to a no downsampling.

The main advantage of using the multi-scale alignment algorithm is to reduce the computation time. To evaluate how much scaling would speed up the algorithm, the elapsed time for

the alignment algorithm is computed for the downsampled images, see Fig. 6-26. Considering the acceptable alignment range of $\mathbf{Scale} = [0.4, 1]$, it can be inferred that the elapsed time decreases from T (sec) to 0.40T (sec) from scale 1 to 0.5. In other words, if an error $\approx$2°is considered acceptable, the image can be downsampled to less than half of its original size to gain a threefold increase in computation speed. $T \approx 10(sec)$ in the Matlab implementation over an Intel(R) Core(TM)2 Duo CUP E7500, with 2.93GHz clock.

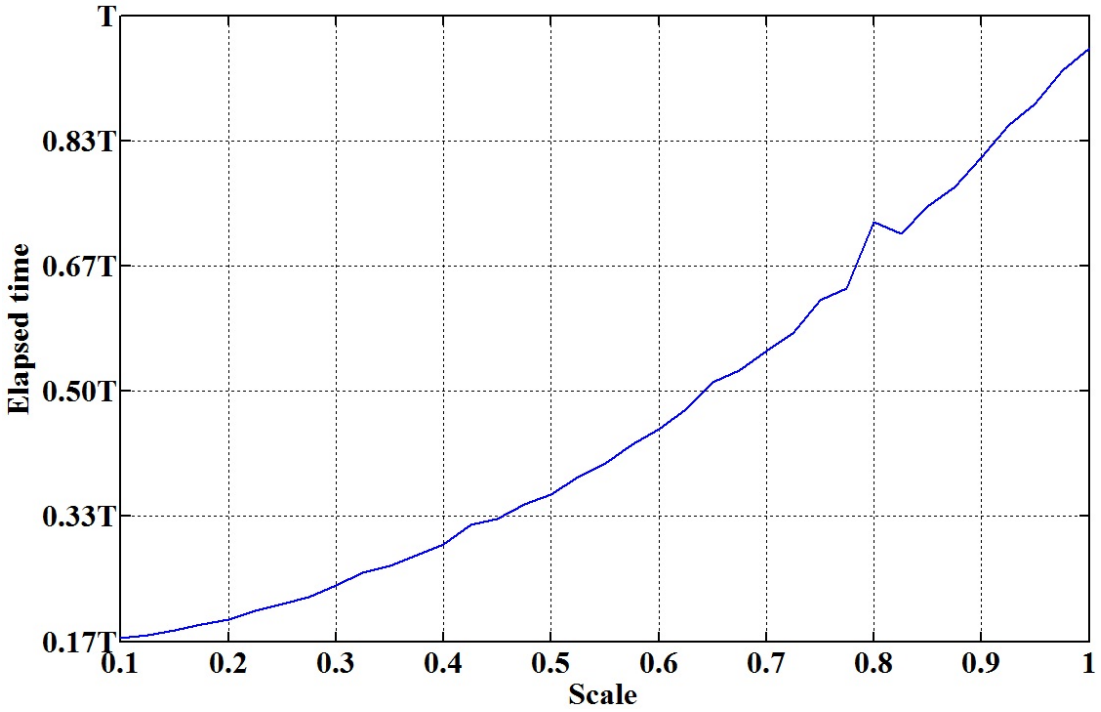


Figure 6-26: The alignment process elapsed time in different scales for 10°, 20°and 30°self-occluded images in the Bosphorus dataset; $\mathbf{Scale} = 1$ corresponds to a 160×100 image. $T \approx 10(sec)$ in the Matlab implementation.

6.5 Conclusion

This chapter extends the application of the IFill(.) operator to aligning self- and partially occluded 3D faces. The algorithm uses two binary operators defined over the nasal region to define objective functions, whose minima correspond to the aligned faces. The algorithm has been compared to the PCA approach, which has been extensively used for 3D face alignment in the literature. The consistency of the algorithm is quantitatively evaluated, using the error generated by the ICP algorithm, over the FRGC dataset. Then, a method to artificially generate

self-occlusion is used to compute the alignment accuracy for the given rotational variations. The same procedure is performed over the Bosphorus dataset's rotated samples.

The major weakness of the proposed algorithm compared to the PCA pose correction approach is the computational speed and convergence. Since the SA used to find the optimal point for the objective function is a non-deterministic algorithm, the convergence is not guaranteed. The algorithm depends on the selection of appropriate values for the initial temperature and maximum number of iterations. Also, the cooling process must be performed very slowly, to provide opportunity for the optimisation algorithm to avoid local minima. These operations consequently increase the computation time. Using other optimisation algorithms or parallelising the SA steps could be some approaches to resolving these problems.

Chapter 7

Nasal curve matching

7.1 Introduction

The nasal region is relatively a stable part on the face and compared to the other regions such as the forehead, eyes, cheeks, and mouth, its structure is comparatively consistent over different expressions [67, 95, 140]. It is also one of the parts of the face that is least prone to occlusions caused by hair and scarves [39]. Indeed, it is very difficult to deliberately occlude the nose region without attracting suspicion [40]. In addition, the unique convex structure of the nasal region makes its detection and segmentation more straightforward than other parts of the face, particularly in 3D.

The nasal region therefore has a number of advantageous properties for use as a biometric. However, it has been suggested that the texture and color information of the 2D nose region does not provide enough discrimination for human authentication [18]. This problem as been ameliorated by the developments in high resolution 3D facial imaging over the last decade, which have led a number of researchers to start studying the potential of the nose region for human authentication and identification. One of the main motivations for this is to overcome the problems posed by variations in expression, which can significantly influence the performance of face recognition algorithms.

This chapter proposes a new recognition technique using the nasal region. Using robustly defined landmarks around the edge of the nose, a collection of curves connecting the landmarks are defined on the nose surface and these form the feature vectors. The approach is termed the Nasal Curves Matching (NCM) algorithm. The algorithm starts by preprocessing the input data. Images are denoised, the face is cropped and then aligned using Mian *et al.*'s iterative

PCA algorithm [32]. Then, the nose region is cropped and a landmarking algorithm used to detect 16 fiducial points around the nose region. Taking the landmarks in pairs, the intersection of orthogonal planes passing through each pair with the face region defines the facial curves. The resulting curves are normalized and used as the feature vectors. Finally, feature selection is used to extract the features that are most robust to variations in expression.

The remainder of this chapter is organized as follows. First, in section 7.2, the preprocessing algorithm is explained. Section 7.3 describes the landmarking algorithm and the construction of the nasal curves, and the feature selection is explained in section 7.4. Using the FRGC v2.0 [14] and Bosphorus [13] datasets, experimental results for the feature selection and classification performance are presented in Section 7.5. Finally, conclusions are drawn in section 7.6.

7.2 Preprocessing

Preprocessing is a vital step in the face recognition systems. Its poor performance can significantly affect the recognition performance and the rest of the algorithm, for example by degrading the feature extraction and the feature's correspondence between samples. Here, a 3 stage preprocessing approach is employed. First, the data is denoised and the face region is cropped. Then, the face is aligned and resampled using a PCA-based pose correction algorithm and, finally, the nose region is cropped.

7.2.1 Denoising, tip detection and face cropping

3D face images are usually degraded by impulsive noise, holes and missing data. Although the noise effects are more salient on the depth $\mathbf{Z}$ information, the $\mathbf{X}$ and $\mathbf{Y}$ coordinates can also be affected. In order to remove the noise in $\mathbf{X}$ the standard deviation of each column is first calculated. Columns with high standard deviations will contain noise while the columns with low standard deviations are relatively noise free. Therefore, the two columns with the lowest standard deviation are found and the $\mathbf{X}$ map's slope found. The slope is then used to resample the map. Using the standard deviation of its rows, the same procedure is used to denoise the $\mathbf{Y}$ map. The $\mathbf{Z}$ map is then resampled using the new $\mathbf{X}$ and $\mathbf{Y}$ maps.

Removing the noise from the depth map is performed by locating the missing points and then replacing them using 2D cubic interpolation. Then, morphological filling is applied to the depth map. Those points whose difference with the filled image is larger than a threshold are assumed to be holes and are again replaced by cubic interpolation. This procedure helps to preserve the natural holes on the face, in particular near the eye's corners. Finally, median

filtering with a $2.3\ mm \times 2.3\ mm$ mask is used to remove the impulsive noise on the face's surface.

The next step is detection of the nose tip. To do this, the principal curvatures (κ_1 and κ_2) are used to find the SI by

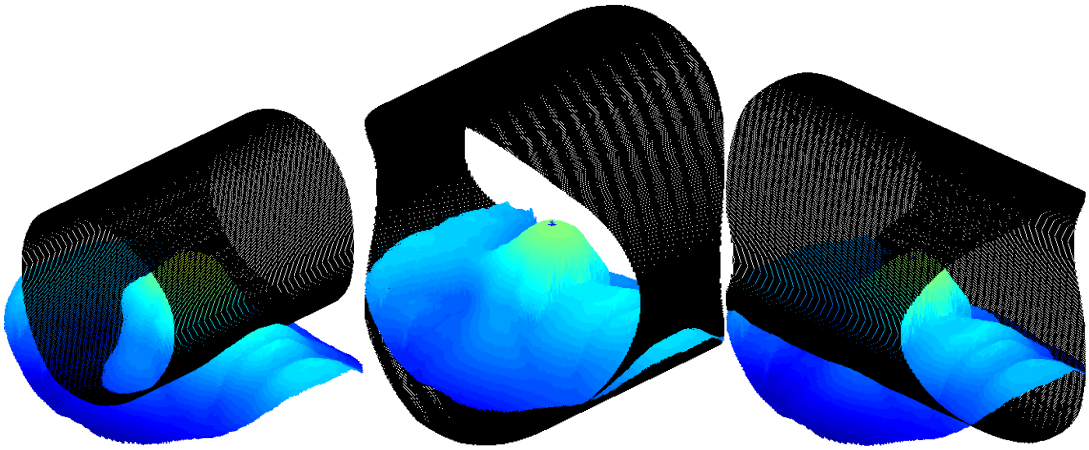
$$\text{SI} = \frac{2}{\pi} \arctan(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}). \tag{7.1}$$

The SI is scaled so that its maximum and minimum values are +1 and -1, respectively, and the face's convex regions found by thresholding the SI to produce a binary image, using -1 $<$ SI $<$ $-\frac{5}{8}$ [67, 140, 39, 40]. The largest connected component is detected, its boundary smoothed by dilating with a disk-shaped structuring element and its centroid saved as the nose tip. Finally, the face region is cropped by intersecting a sphere of radius 80 mm, centered on the nose tip, with the face.

7.2.2 Alignment and nose cropping

The face region is aligned using the PCA based alignment algorithm of Mian *et al.* [32]. In each iteration of the algorithm resampling is performed with resolution of 0.5 mm, self-occluded points are replaced by 2D cubic interpolation and the nose tip is re-detected. To do this, after resampling, the SI is again calculated and the biggest convex region is located. The face is again cropped and PCA is applied on the newly cropped image. This simple process helps to localize the tip more accurately. After the alignment procedure is completed a small constant angular rotation along the pitch direction is added to the face pose, as this helps the landmarking algorithm to detect the nose root (radix).

The nose region is cropped by finding the intersections of three cylinders, each centered on the nose tip, with the face region. Two horizontal cylinders, with radii 40 mm (Fig. 7-1-a) and 70 mm (Fig. 7-1-b), crop the lower and upper parts of the nose, respectively. Then, a vertical cylinder, of radius 50 mm (Fig. 7-1-c), bounds the nose region on the left and right sides. Applying these conditions over the **X**, **Y** and **Z** maps results in a binary image, which is further trimmed by morphological filling and convex hull calculation. The final binary image is used to find the cropped nose point clouds. This approach to nose region cropping results in fewer redundant regions than the approach of [39] and is much faster than that of [95], which uses level set based contours.



(a) Lower parts (b) Upper parts (c) Left and right parts

Figure 7-1: The cropped face and cylinders intersections.

7.3 Nasal region landmarking and curves finding

The sixteen landmarks that are detected on the nose region are shown in Fig. 7-2. A cascade algorithm is used to directly find the nose tip (**L9**), root (**L1**), and the left (**L5**) and right (**L13**) extremities. First, **L9** is detected and then used to detect **L1**, **L5** and **L13**. To avoid selecting incorrect points resulting from residual noise or the nostrils as landmarks, an outlier removal procedure is employed and this procedure is explained in Section 7.3.4. The remainder of the landmarks are found by sub-dividing lines connecting the landmarks already found. In the following subsections the landmarking approach is explained in detail.

7.3.1 Nose tip L9 detection

Although the nose tip has already been approximately localized, it is more accurately fixed in this step. The SI is again thresholded to extract the largest convex regions from the cropped nose region. Then, the nose region's depth map, $\mathbf{Z_n}$, is inverted and the largest connected region is located [39]. The resulting binary image is multiplied by the convex region to refine it and remove noisy regions. The result is dilated with a disk structuring element and multiplied by $\mathbf{Z_n}$. After median filtering the result, the maximum point is considered as the nose tip. The reason for not directly selecting the maximum point of $\mathbf{Z_n}$ as the tip is its vulnerability to residual spike noise.

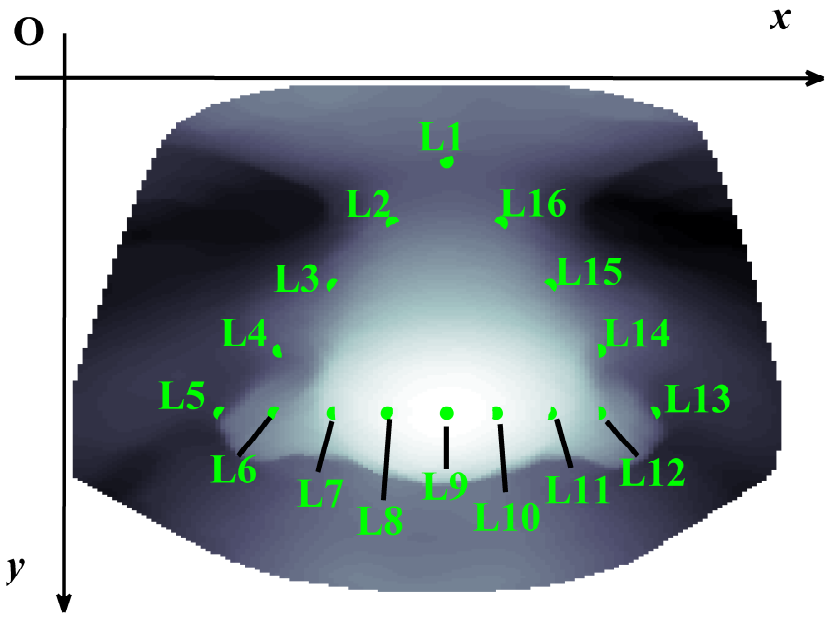


Figure 7-2: The locations of the landmarks and their names.

7.3.2 L1 detection

A set of planes perpendicular to the *xy* plane and containing **L9** are then found, as shown in Fig. 7-3. The angle between the i^{th} plane and the *y*-axis is denoted as α_i with a normal vector given by $[\cos\alpha_i, \sin\alpha_i, 0]$. Intersecting the nose surface and the planes results in a set of curves. The global minimum of each curve's depth is found and the landmark **L1** is located at the maximum of the minima. This procedure is depicted in Fig. 7-3, where $\alpha = 15°$is the maximum value of $[\alpha_1, \alpha_2, \ldots, \alpha_M]$.

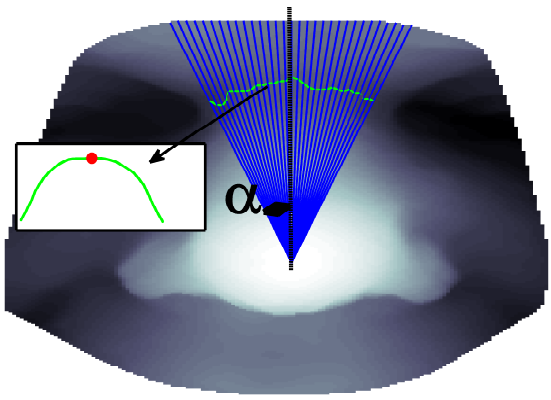


Figure 7-3: **L1** detection procedure: the blue lines are the planes' intersections, the green curve is each intersection's minimum and **L1** is given by the maximum value of the minima, shown with a red dot.

7.3.3 Detection of L5, L13 and the remaining landmarks

A set of planes which include **L9**, are perpendicular to the *xy* plane and have the angular deviation β_i with the *x*-axis are intersected with the nose surface (Fig. 7-4). The normal of each planes is given by $[\sin\beta_i, \cos\beta_i, 0]$ and the intersection of the planes and the nose surface results in a set of curves ($i = 1, \ldots, N$). **L5** and **L13** are located at the peak position of the curves' gradient. To find these, each curve is differentiated and the locations of the peak values stored. This results in a set of points on the sides of nasal alar. After outlier removal, as described in the next section, the points with the minimum vertical distance from the nose tip (**L9**) are chosen as **L5** and **L13**.

After the four key landmarks were detected, they are projected on the *xy* plane. The lines connecting the projection of **L1** to **L5**, **L5** to **L9**, **L9** to **L13** and **L13** to **L1** are divided into four equal segments and the *x* and *y* positions of the resulting points are found. The corresponding points on the nose surface give the remaining landmark locations, shown in Fig. 7-1.

7.3.4 Removal of outlying landmark candidates

As the candidate positions for the landmarks **L5** and **L13** are the positions of maximum gradient on the nose surface, they are sensitive to noise and the position of the nostrils. In order to remove incorrect candidate positions an outlier removal algorithm is proposed. With reference to Fig. 7-4, the gradient maxima of the intersection of the planes with the nose surface are marked as green points. However, some outliers are detected as candidates for **L5**, in this case due to the impulsive depth change around the nose tip (located within the black circle in Fig. 7-4). To remove, the outliers the distances from the candidate points to the nose tip are clustered using K-means with $K = 2$. The smallest cluster will contain the outliers and these points are then replaced by peaks in the surface gradient that are closer to the centroid of the larger cluster. The replacement candidates are plotted in red in Fig. 7-4.

A similar gradient-based approach for detecting the side nasal landmarks was proposed in [44], where the locations of the peaks of the gradient on the intersection of a horizontal plane passing through the tip and the nose surface were selected as **L5** and **L13**. However, by using a set of candidates instead of just a pair and the outlier removal, the approach proposed above is more robust.

7.3.5 Creating the nasal curves

After translating the origin of the 3D data to the nose tip, the landmarks are used to define a set of nasal curves that form the feature space for each nose. Considering any two pairs

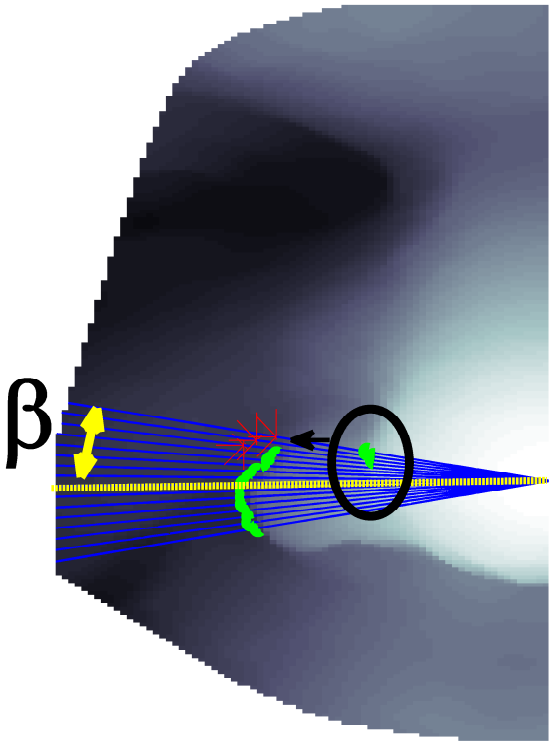
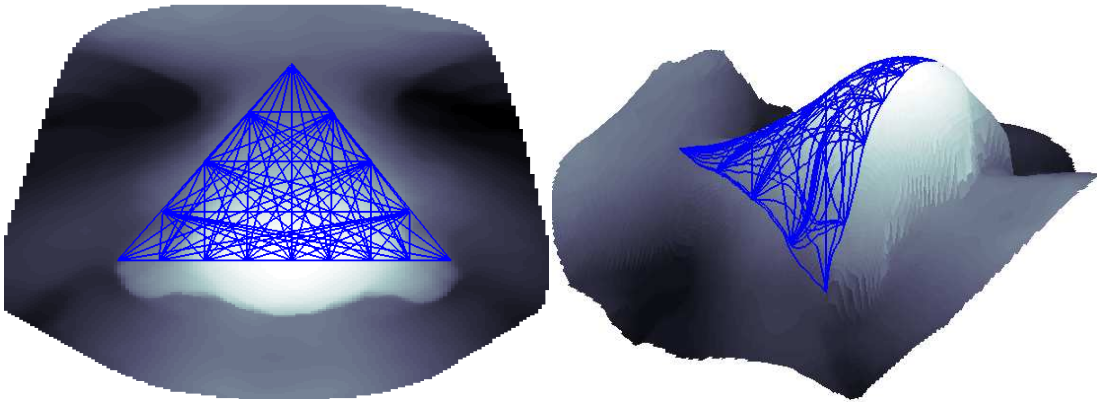


Figure 7-4: **L5** (and similarly **L13**) detection procedure: intersections of the orthogonal planes (blue lines); candidate points for **L5** (green points) and outlier removal results (red points). $\beta = 15°$is the maximum of $[\beta_1, \beta_2, \ldots, \beta_N]$.

of landmarks, the intersection of a plane passing through the landmarks and perpendicular to the *xy* plane with the nose surface can be found. The normal vector of the plane is given by $\frac{(\mathbf{L_i}-\mathbf{L_k})\times\hat{a}_z}{|(\mathbf{L_i}-\mathbf{L_k})\times\hat{a}_z|}$, where $\mathbf{L_i}$ and $\mathbf{L_k}$ are the two landmarks and $\hat{a}_z$ is the unit vector of the *xy* plane. The 75 nasal curves depicted in Fig. 7-5 are found by connecting the following landmark pairs:

1. **L1** to **L2**-**L8** and **L10**-**L16**;
2. **L2** to **L6**-**L8** and **L10**-**L16**;
3. **L3** to **L16**, **L10** -**L15** and **L6**-**L8**;
4. **L4** to **L14**-**L16**, **L10**-**L13** and **L6**-**L8**;
5. **L5** to **L13**, and **L6**-**L7**;
6. **L9** to **L1**-**L5** and **L13**-**L16**;
7. **L14** to **L5**-**L8**, **L10**-**L12**;
8. **L15** to **L5**-**L8**, **L10**-**L12**;
9. **L16** to **L5**-**L8**, **L10**-**L12**.

Each curve is then resized to a fixed length and their maximum depth translated to zero. The points from the complete set of 75 curves form the feature vector and are used for nose recognition.



(a) (b)

Figure 7-5: The nasal curves: (a) Frontal and (b) side view.

7.4 Expression robust feature selection

The set of nasal curves shown in Fig. 7-5 provide a fairly comprehensive coverage of the nasal surface. However, simply concatenating the curves produces a high dimensional feature vector that will typically suffer from the curse of dimensionality and so not produce the best classification performance. In addition, some of the curves are intrinsically more sensitive to the deformations caused by expression while others may be affected by the wearing of glasses, one of the most common occlusions found in biometrics sessions. Therefore, it is desirable to select a subset of curves that produce the best recognition performance over a range of expression variations and occlusions from glasses. By considering each curve as a set of features that are either included or excluded from the feature vector, the nasal curves that contribute to a robust recognition performance can be investigated. To do this, the well-known Forward Sequential Feature Selection (FSFS) algorithm is employed. Using FSFS, the single curve that produces the best recognition performance is found and then additional curves are iteratively added to form the set of the best n features. The cost function used to evaluate the recognition performance is

$$E = R_1. \tag{7.2}$$

where R_1 is the rank-one recognition rate. The ranks are obtained using the leave-one-out approach and nearest neighbor city-block (CB) distance calculation.

7.5 Experimental results

To evaluate the performance of the NCM algorithm the Bosphorus and FRGC v2.0 datasets are utilized for feature selection and matching, respectively. Two matching scenarios are used and the sensitivity to the number of training samples is analyzed.

The Bosphorus dataset includes a set of frontal samples having various expressions: neutral, happy, surprise, fear, sadness, anger and disgust. These samples are used below to select the most expression invariant nasal curves.

7.5.1 Landmarking and feature selection results

Finding the ground truth for the landmarks in 3D is very difficult. The landmarking algorithm's accuracy can be verified by the recognition results, discussed later. However, its consistency can be evaluated by translating the nose tip (**L9**) to the origin and then, for each subject, calculating the Euclidean distance between each landmark in all captures. For all subjects, the mean and standard deviation distances (in mm) for **L1**, **L5** and **L13** are 3.3620 ± 1.6798, 2.3557 ± 1.1822 and 2.4259 ± 1.2691, respectively. These results are very competitive with those recently reported in [69] which, unlike the approach presented here, require training.

Feature selection is performed using FSFS and evaluated using the Bosphorus dataset. In all experiments, the facial curves were resampled to a fixed size of 50 points and concatenated to create the feature vector. Using a fixed number of points was found to produce a higher recognition performance than varying the number of points per curve according to the curves' length and the performance was also relatively insensitive to the number of points per curve.

Figure 7-6 plots the rank-one recognition rate against the number of nasal curves in the feature set and also illustrates the curves selected for a number of points on the plot. For example, the first curve selected is that connecting **L1** to **L9** (**L1L9**), then **L4L13** is selected giving the combination of **L1L9** and **L4L13**.

Overall, the highest rank-one recognition rate occurs when 28 curves are selected. The distribution of these curves, shown in Fig. 7-6, is relatively even over the nasal surface but is slightly denser on the nasal cartilage, which is less flexible due to its bony structure, and on the alar. After this, the rank-one recognition rate decreases as more features are added which conforms with expectations. The 28 curves, in order of FSFS selection, are: **L9L1**, **L4L13**, **L5L13**, **L1L4**, **L15L5**, **L2L13**, **L1L14**, **L2L12**, **L3L6**, **L1L7**, **L9L5**, **L1L2**, **L16L8**, **L9L13**, **L3L16**, **L1L16**, **L16L5**, **L1L10**, **L16L6**, **L15L7**, **L16L12**, **L15L8**, **L14L12**, **L14L5**, **L1L5**, **L9L2**, **L15L11** and **L3L12**. As these curves produce the best recognition performance for a dataset with a wide range of expressions, they should be relatively insensitive to variations in expression.

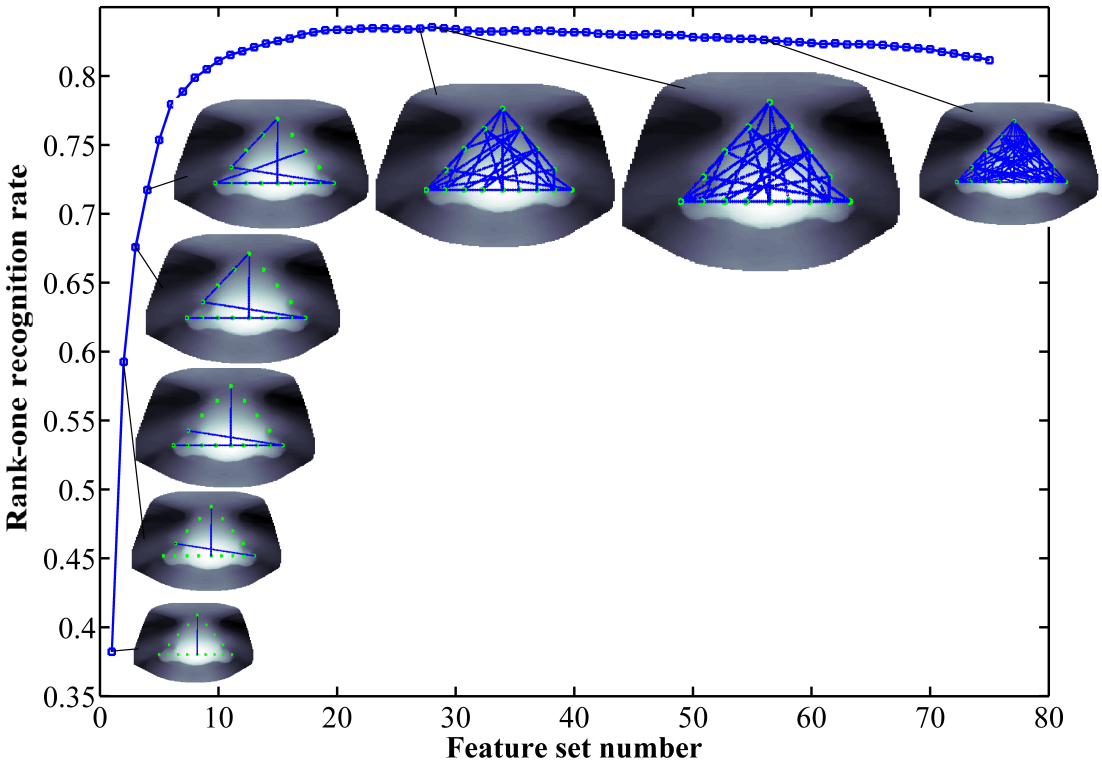


Figure 7-6: Rank-one recognition rate against the number of nasal curves selected by the FSFS algorithm. The sets of curves for selected feature sets are also shown, with the largest image (second from right) showing the 28 curves that produced the highest recognition rate.

For comparison, a GA is also used to select the best performing feature sets. Equation (7.2) is used as a measure of fitness and a 75 dimensional parameter space to represent the curves. Compared to FSFS, which is a deterministic algorithm, GA stochastically maximizes the rank-one recognition rate. Although GA have the capability to examine various combinations of the features their convergence is not guaranteed. Cumulative Match Characteristic (CMC) recognition results for the best performing sets of curves selected by GA and FSFS are plotted in Fig. 7-7. The FSFS curves outperform those selected by GA in terms of recognition, computational speed and convergence. In addition, while the best performing FSFS set had only 28 curves, the GA set contained 33 curves.

7.5.2 Classification performance

The recognition performance of the NCM algorithm is evaluated using FRGC v2.0 dataset. In the experiments, the feature vectors are formed by concatenating the 28 expression robust curves found by FSFS on the Bosphorus dataset, see Fig. 7-6. As before, all curves are resampled to 50 points and each curve is normalized by translating its maximum to zero.

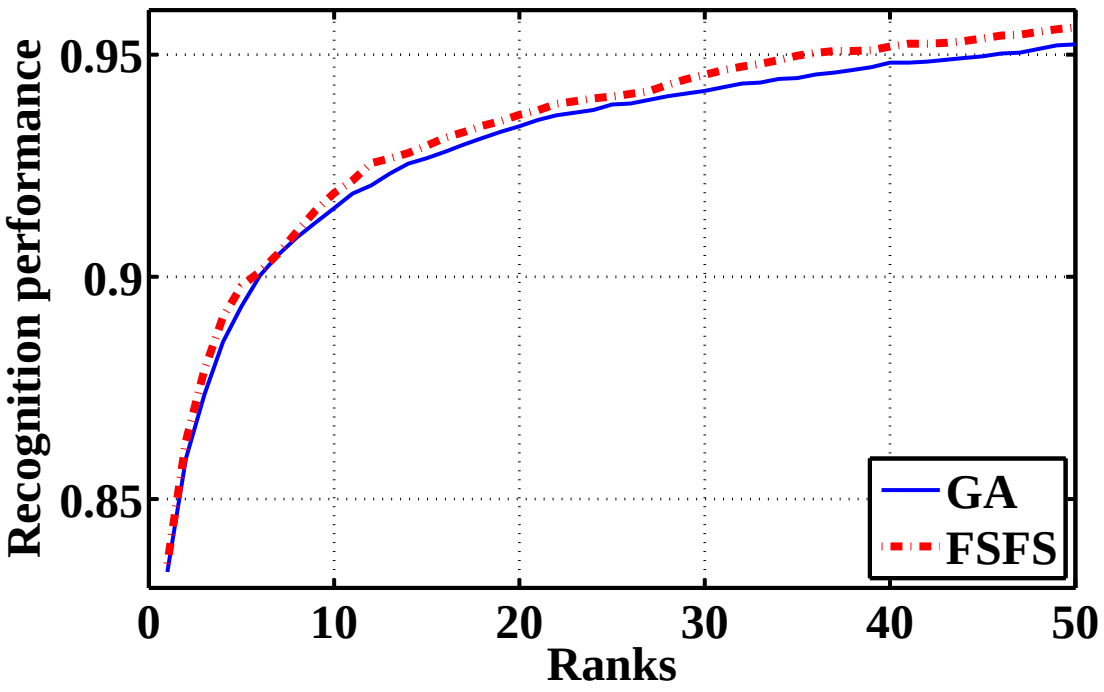


Figure 7-7: Cumulative match characteristic (CMC) curve for the best feature sets found by FSFS and GA feature selection results.

Two scenarios are used to evaluate the NCM algorithm. The first one is the all-vs.-all scenario, in which all of the folders in the FRGC v2.0 dataset are merged. From the merged folders 505 subjects with at least two samples are selected giving a total of 4879 samples. The number of training samples per class is varied from 1 to 12 and the rank-one classification performance of a variety of classification methods found. The classification methods used are PCA, linear discriminant analysis (LDA), Kernel-Fisher's analysis (KFA), direct CB distance calculation, multi-class SVM (Multi-SVM) and bootstrap aggregating decision trees (TreeBagger). The PCA, LDA and KFA algorithms were implemented using the PhD toolbox (Pretty helpful Development functions for face recognition), while the Matlab's Statistics Toolbox is used for the SVM and TreeBagger classification. For the subspace classification methods, the final matching is performed using the nearest neighbor CB distance calculation.

Figure 7-8 shows the rank-one recognition results for the all-vs.-all scenario. Matching using the direct calculation of the CB distance produces the worst recognition performance for ≥ 6 training samples. LDA and PCA project the feature space to a 277-dimensional subspace. These methods require a sufficient number of training samples per class to be trained appropriately [131] and for low numbers of training samples PCA fails to find the direction with the highest variance properly. This problem is more severe for LDA and is reflected in the low recognition rate for ≤ 5 training samples. However, as the number of training samples

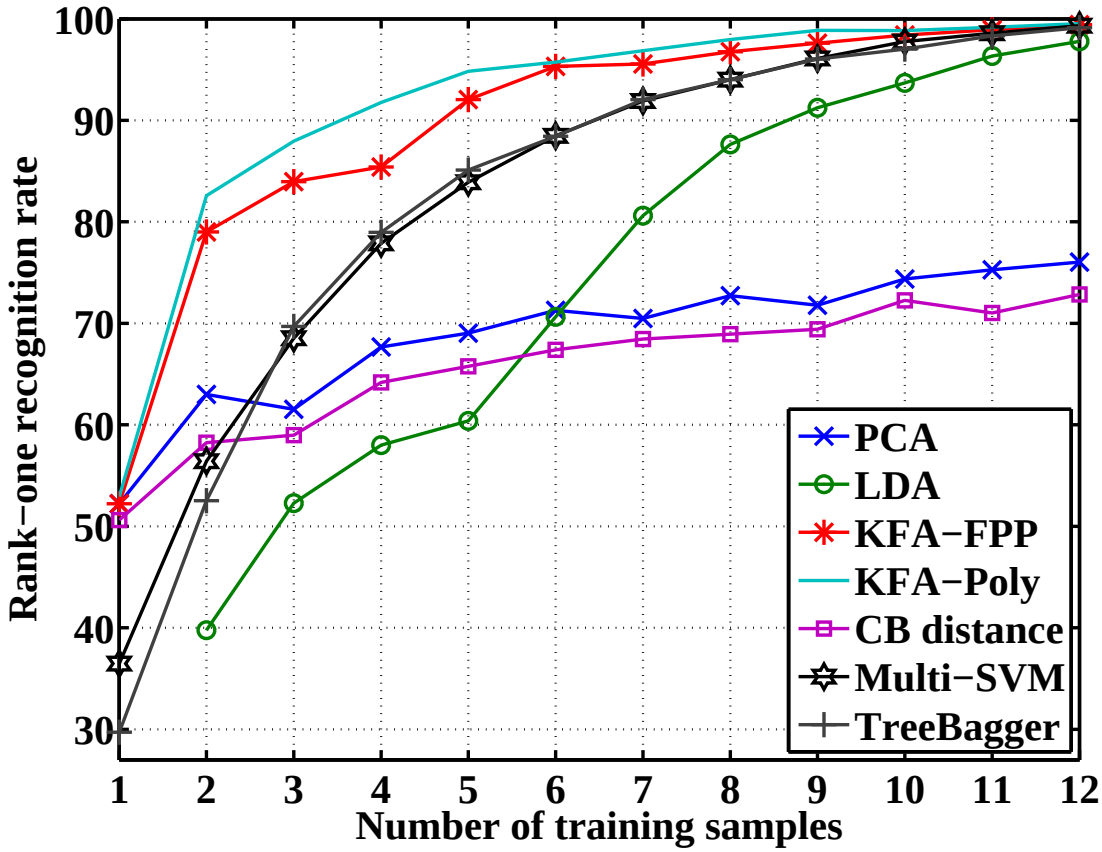


Figure 7-8: The rank-one recognition results using different numbers of training samples and classification methods.

increases, the classification performance of these subspace projection techniques improves, in particular for LDA whose peak recognition rate reaches 97.78% for 12 training samples. To implement the multi-SVM classifier [156] the one-vs.-all scenario is used to generalize SVM to a multi-class classifier. Again, for low numbers of iterations the recognition performance is low but dramatically increases with the number of training samples, up to 99.32% for 12 training samples. The TreeBagger classifier has the same trend, rising form a low rank-one recognition rate for a single training sample to 99.13% for 12 training samples. An ensemble of 119 trees are aggregated for the tree classifier.

The issue of low training samples per class can be addressed by using kernels for Fisher's analysis [157]. Two kernels are used, the fractional power polynomial (FPP) kernel [84] and the polynomial (Poly) kernel. Figure 7-8 shows that both kernels result in a significant improvement in the recognition performance, with rank-one rates of 82.58% and 79.01% for the Poly and FPP kernels, respectively, increasing to 99.5% for both kernels using 12 training samples.

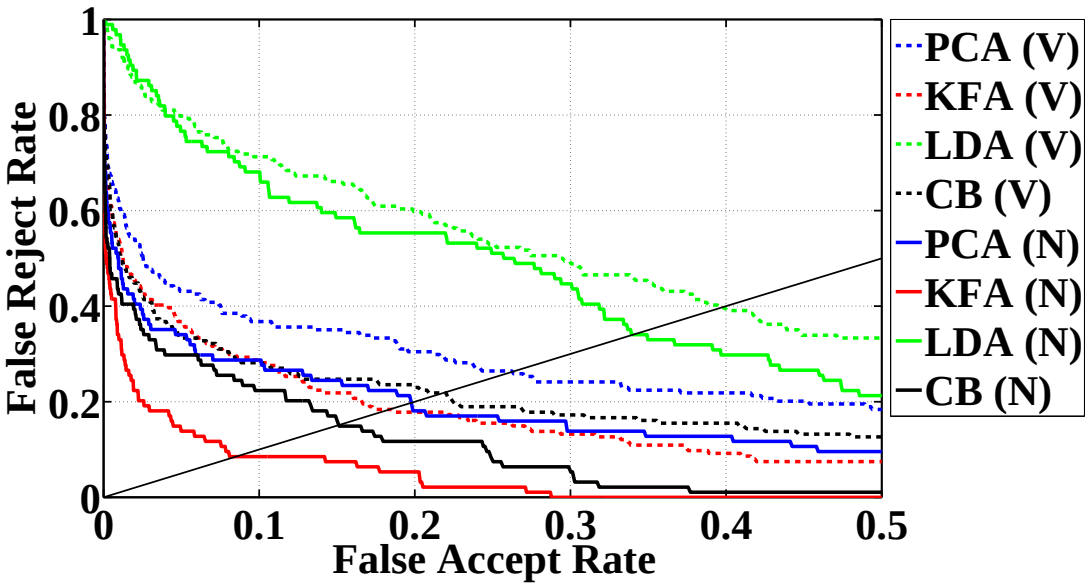


Figure 7-9: ROC curves for the neutral (dashed line - N) and varying (solid line - V) expression samples: anger, smile, surprised, disgust and open mouth.

The second scenario is based on the FRGC Experiment 3 [14]. The 3D samples in the Spring2003 folder, consisting of 943 samples from 275 subjects, are used for training and the other two folders are used for validation (4007 samples). The only difference with the original Experiment 3 is that, as the NCM algorithm only uses the 3D information, no color or texture information is used. Two different experiments are performed: one using the neutral faces Fall2003 and Spring2004 folders as the probes and the other using the non-neutral samples in the probe folders. The receiver operating characteristic (ROC) curves and equal error rates (EER) for both neutral and non-neutral probes are given in Fig. 7-9. For the neutral probe images, KFA-Poly again produces the best verification rate, with an EER of 0.08, while, LDA produces the poorest verification, again due to its sensitivity to few number of training samples.

When the non-neutral samples are used, the EER increases for all classification techniques. KFA-Poly still has the lowest EER at 0.18. Just above is the CB distance nearest neighbor classifier which performs better than PCA and LDA in this case. One reason for this could be that the CB distance has better discriminatory power when the feature space is sparse as it uses the L1-norm [130]. For comparison, the EER for the best performing combination of two nasal regions from [67] are provided, see Table 7.1. This work used the ICP algorithm for matching and the same dataset for verification. The EER for the NCM algorithm using the KFA-Poly classifier are 0.04 and 0.05 below those of [67] for neutral and varying probes, respectively.

The final evaluation in Table 7.2 compares the rank-one recognition rates achieved by the NCM algorithm with other nose region recognition results reported in the literature for probe3s

Algorithm	Matching	Expression	
		Neutral	Varying
NCM	KFA-Poly	0.08	0.18
Chang *et al.* [67]	ICP	0.12	0.23

Table 7.1: Comparison of EER for the best performing KFA-Poly curve from Fig. 7-9.

with neutral and varying expressions [67, 140, 95, 89].

From [67], the best performing single probe is used for comparison. This probe contained the complete nasal region and some surrounds. One-to-one matching was performed using ICP which, although training free, can be relatively expensive. Although the NCM KFA-Poly results were $\approx 6\%$ lower than [67] for neutral expressions, for varying expressions the difference was only $\approx 1\%$ which is encouraging given the low complexity of the simple 1D curves used in the matching.

For the Bosphorus dataset, following Dibeklioğlu *et al.* [140], the leave-one-out algorithm is used and the algorithms are applied to the frontal view samples with various expressions. The average rank-one for the best NCM result is 3.34% higher than that reported by [140], which again used ICP for the 3D matching .

Both [95] and [89] used a gallery containing both neutral and varying expressions. Despite the small FRGC subset of 125 subjects used for evaluation, the nose contours matching method of Drira *et al.* [95] only produced a rank-one recognition rate of 77%. Wang *et al.* [89] used a local shape difference boosting method and Haar-like features from regions cropped by spheres centred on the nose tip. When the sphere's radius $r = 44$ mm the cropped region

Reference	Dataset	Expression	
		Neutral	Varying
Chang *et al.* [67]	FRGC	96.6% (ICP)	82.7% (ICP)
NCM	FRGC	90.87%	81.61%
	Bosphorus	97.44%	
Dibeklioğlu *et al.* [140]	Bosphorus	94.1% (ICP)	
Drira *et al.* [95]	FRGC subset	77% (Geodesic contours)	
Wang *et al.* [89]	FRGC	95% (radius = 44 mm) 78% (radius = 24 mm)	
NCM	FRGC	89.61%	

Table 7.2: A comparison of the rank-one recognition rate for NCM to other recently reported nasal region recognition techniques.

includes parts of the cheeks and mouth in addition to the nose and the rank-one recognition rate is approximately 95%. However, when $r = 24$ mm only the nose tip is cropped and the rank-one recognition rate drops to 78%. The NCM result of 89.6%, is consistent with these, considering its features are strictly contained on the nasal surface. In addition, [89] used 3541 FRGC samples, approximately 500 fewer than for the NCM results.

7.6 Summary and discussion

A new 3D nose recognition algorithm is proposed. The motivation for using the nose region is its relative invariance to variations to expressions. At the heart of the method is a robust, training-free landmarking algorithm which is used to define 16 landmarks on the nose surface around the edge of the nose. By taking the landmarks in pairs, a set of 75 curves on the nose surface are generated and these form the feature set for matching. FSFS is applied to the feature set and 28 nasal curves that produce a robust performance over varying expressions is selected. These curves are used for authentication and verification scenarios, using a selection of classification algorithms. Results obtained from the FRGC v2.0 and Bosphorus 3D face datasets show a good classification performance with low EER. Comparison with other reported recognition results for the nasal region shows a competitive classification performance, particularly for varying expressions.

The classification results demonstrated that the KFA-Poly classifier is able to overcome the problem posed by low numbers of training samples per subject. On the other hand, other subspace-based classifier such as PCA and LDA fail to find appropriate projections when few training samples are available. Multi-SVM and TreeBagger have much better performances when the number of training samples per class increases. In the experiments in this work, linear kernels and one-vs.-all scenario are used for Multi-SVM. However, using non-linear kernels and other multi-classification scenarios has the potential to address the low training samples issue and hence increase the SVM's classification performance.

The current work can be extended in many aspects. Fusing the proposed feature space with holistic facial features such as depth, Gabor wavelets or LBP has the potential for increasing the recognition performance. In addition, the NCM algorithm could also be used as a robust pattern rejector, to robustly reject many faces after computing their nasal region recognition ranks, hence reducing the complexity when very large datasets are used.

Chapter 8

Expression robust nasal region recognition

8.1 Introduction

The nasal curves matching (NCM) algorithm proposed in the previous chapter computes nasal curves over the depth map to create the feature space. Although it shows a high robustness against deformations caused by expression variations, it also has some deficiencies which are explained in this Chapter and significantly improved using a new nasal region recognition algorithm. The main issues with the NCM algorithm can be categorised as follows,

1. Feature space: the depth is used as the basis of the feature space in the NCM algorithm. Although it is easy to be utilised and includes significant information about the facial surface, it is vulnerable to noise and alignment errors. In addition, small pose variations can reduce the landmarking algorithm's accuracy and degrade the between-class dissimilarity of the feature space.

2. Landmarking: the NCM landmarking approach is very consistent and training-free. However, it was observed that for some faces the minimum computation, which is the foundation of the landmarking algorithm, fails to detect a minimum. This is because of the nature of some facial curvature, which is either strictly descending or have multiple minima which is usually caused by hair or facial expressions. As a consequence, the

sign of the differentiated surface is not changed and no minima is detected. Also, the K-means algorithm used for outlier removal might generate empty clusters. K-means is non-deterministic and depending on the location of initial centroids, can produce different results every time it is run.

3. Feature descriptors: nasal curves are utilised in the NCM algorithm as its feature descriptor. However, some curves are not necessarily significantly different between different subjects (cross-class dissimilarity) and are vulnerable to landmarking position. Although this issue might be resolved by additional nasal curves (which might be different among subjects), it can still result in misclassification of samples. On the other hand, since a curve is only able to accumulate very localised surface curvature information, it might fail to obtain robust facial regions for expression deformations and noise. As a consequence, the same two curves for the same subject might be significantly different from each other and within-class scatter would be less concentrated.

4. Feature selection: the FSFS approach is used for feature selection by the NCM algorithm. Although the main reason for this method is its higher computational speed compared to heuristic methods, it is not able to evaluate a high number of combinatorial feature sets. Therefore, some higher discriminant feature sets might not be tested and the feature selection will not be optimal. Also, the one vs. all scenario for calculating matching scores is computationally costly. Also, since it uses the city block distance for matching, it can not optimally discriminant classes with non-linear class discrimination.

To address all these issues, in this chapter a novel method is proposed for nasal region recognition. It is based on a highly consistent and accurate landmarking algorithm. The landmarks are the nose tip, eye corners, nasal root, subnasale and alar groove. Then a feature space based on normals of Gabor-wavelet filtered images are calculated. Instead of just using curves, which have the aforementioned problems, spherical and triangular patches are also applied as feature descriptors. GA is finally used as the feature set selector, using a quick implementation of Mahalanobis-cosine distance, over KFA subspaces. The result is the highest nasal region recognition ranks compared to the previous researches: 1) 97.87% rank-one rate and 2.4% equal error rate for FRGC v2.0 and ROC III experiments, respectively; 98.45% and 98.51% for FRGC's neutral vs. neutral and neutral vs. non-neutral sample, respectively; 3) 96.19% when one gallery sample per subject is used for the FRGC dataset (482 gallery samples (subjects) vs. 4330 probe samples); 4) 97.35% for Bosphorus dataset's neutral vs. non-neutral and "action unit" samples; 5) 95.28% for 105 gallery samples vs. 2797 probe samples for the Bosphorus dataset.

The chapter is organised as follows. first the landmarking algorithm is illustrated in section 8.2. Then an overview of the possible feature types, which can be extracted from the nasal

surface are outlined in section 8.3. The feature descriptor methods and the feature selection algorithm are explained in section 8.4, 8.5, respectively. Experiments and their results are demonstrated in section 8.6. Finally the chapter is concluded in section 8.8

8.2 Landmarking

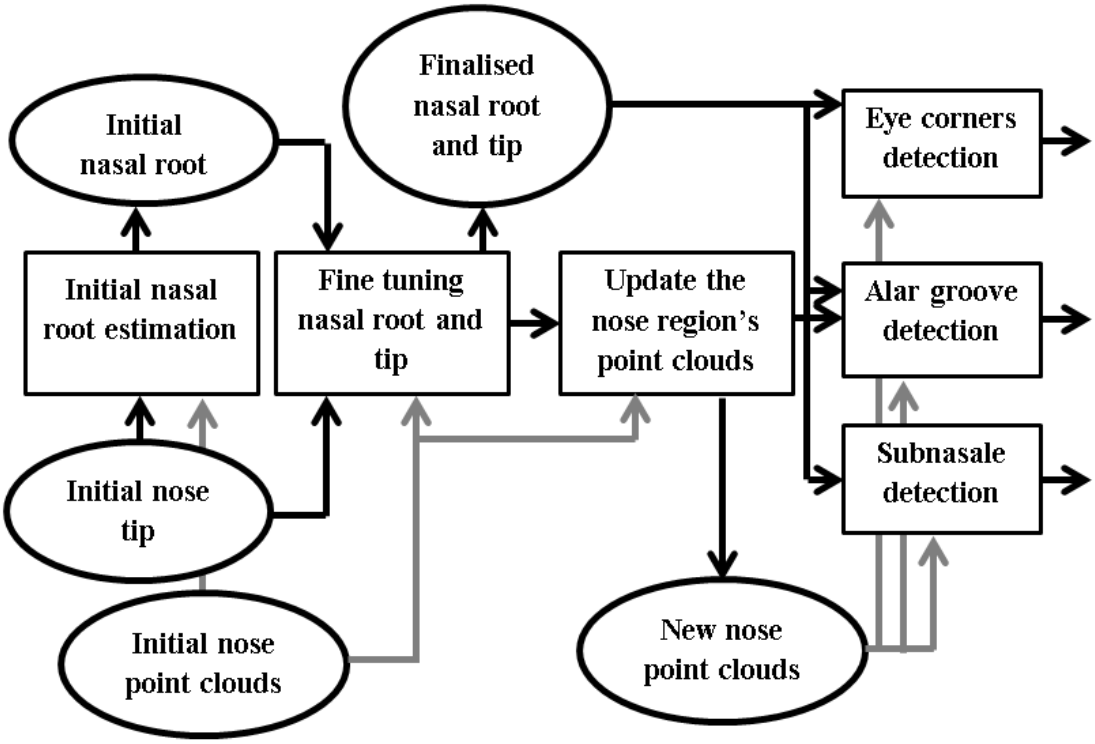
The proposed landmarking algorithm is a cascade approach, which is based on utilising the curvature information and restricting the region of interest (ROI) at every step. The approach is completely training free and as will be discussed in section 8.6.1, is highly consistent in different expressions. The block diagram in Fig. 8-1-a shows how the landmarking is performed. First the more accurate location of the nose tip and root are detected. Then, the nose allar groove, eye corners and subnasale are localised using a parallel approach. The landmarks location and their naming convention used throughout this chapter are plotted in Fig. 8-1-b. The algorithms are detailed in the following subsections.

8.2.1 Minimum detector

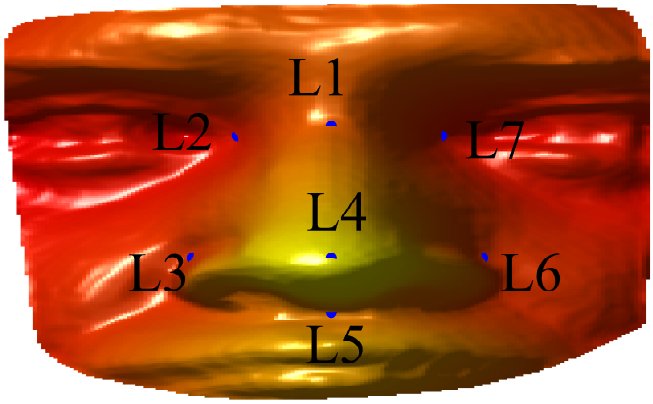
Before explaining how the other landmarks are detected, in this section, a minimum detector function is described, which is utilised in the landmarking process. In some previous landmarking algorithms [44], to be able to localise important landmark on the face, differentiation or gradient operators were used. Although differentiation is indeed helpful to signify curvature variations, it is very sensitive to the noise and deformations on the face surface due to expressions. Also, for some subjects in dataset, the facial surface might not necessarily provide an extremum on the face for the landmarks. The depth around the landmarks might be strictly decreasing or increasing and consequently, no extrema can be detected. The outcome of these instances might be inaccuracy and inconsistency of the landmarking, which adds outliers to the feature space and degrades the recognition performance.

To be able to resolve these issues, instead of finding a single minimum by directly using the depth map, a set of minima are detected from the rotated version of the curve (the same procedure can be also applied to find a set of maxima). Let us assume $\mathbf{P} = [\mathbf{X_f}, \mathbf{Y_f}]$ (a $K \times 2$ matrix) as a set of K points, which represent a curve in the discrete space, and the location ofa minimum on it is sought for. An example of such a curve is depicted in Fig. 8-2. This curve can be obtained by intersecting a plane with the nose surface to pass through the nose alar.

If differentiation operators (or difference operators in the discrete space) are applied over the curve, there will not be any sign change and therefore, no minimum is detected. This is clearly because of the fact that the curve is monotonically decreasing. Instead of applying the



(a)



(b)

Figure 8-1: (a) The landmarking algorithm demonstrated in a block diagram; (b) The nasal landmarks used in this work.

differentiation directly on $\mathbf{P}$, it is first rotated around the z-axis using a given angle α and around a point on the curve (for instance a nasal landmark; In the current figure the point its the curve's origin). Then the local minima of the resulting curve are detected by checking the sign of the differentiation. This operation is mentioned in the following equation,

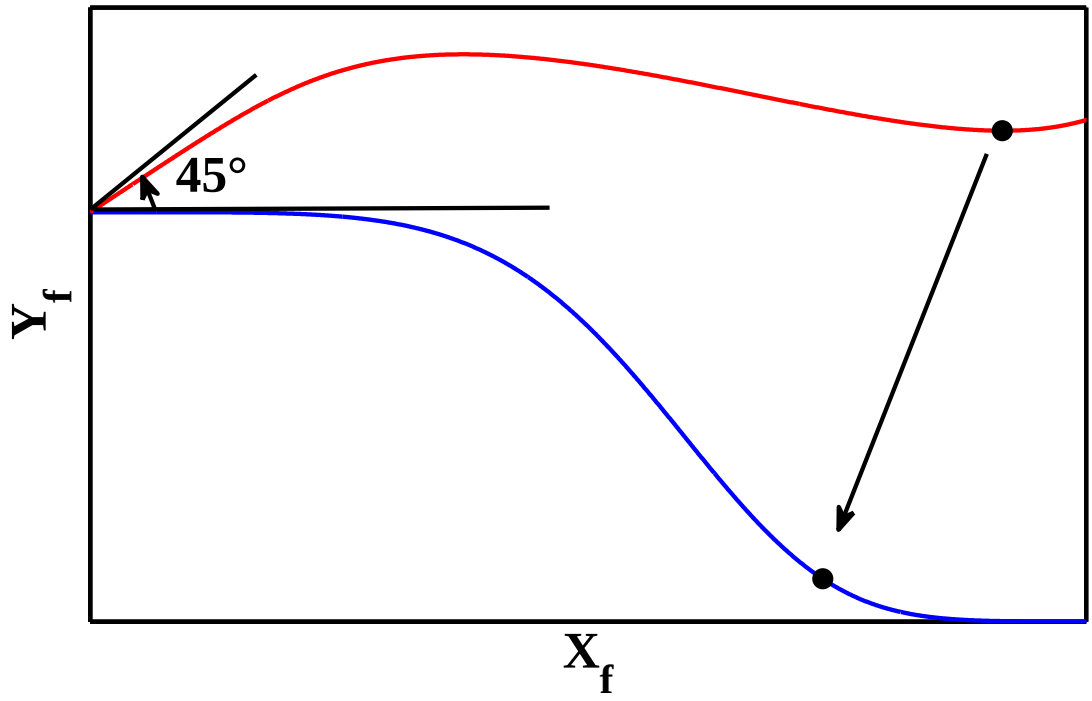


Figure 8-2: The blue curve is an strictly decreasing curve and the red curve is its rotated version (for 45 °as an example). Although the blue curve does not contain any minimum, its rotated version has an obvious minimum. This operation is performed using the $V_n(.)$ function.

$$\begin{cases} \mathbf{P}_\alpha = \mathbf{P} \times \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \\ \mathbf{MIN} = V_n(\mathbf{P}_\alpha) \end{cases} \tag{8.1}$$

in which **MIN** is an $n \times 2$ matrix, including the location of the n lowest valleys (minima) on $\mathbf{P}_\alpha$, which can be easily transform back to the $\mathbf{P}$'s domain. $V_n(.)$ is the valley detector function that computes n lowest clusters of valleys located on the input curve. The advantage of using the rotation is shown in Fig. 8-2. Although the original curve $\mathbf{P}$ does not have a minimum due to its monotonic decreasing trend, its rotated version ($\mathbf{P}_\alpha$) has an obvious minimum.

The minimum detector explained here is used in the following subsections for the landmarks localisation. The proposed landmarking algorithm is mainly based on planes intersection with the nasal region, which results in curves. The minimum detector helps to localise a set of extremma on the curves.

8.2.2 Accurate nose tip re-localisation, nasal root and subnasale detection

The nose tip location can still be more finely tuned. This is performed to increase the consistency of the landmark's location, which is crucial for maintaining the feature space's within-class similarity. Prior to this tuning, an initial position for the nasal root ($\mathbf{L1^0}$) is detected. First, the nasal region is rotated along the pitch direction and around $\mathbf{L4^0}$, to make an artificial minimum for the minimum detector in (8.1). This is perform to avoid minima detection failure for those subjects, whose depth has a strictly decreasing trend in the nasal root region. Then various planes, passing $\mathbf{L4^0}$ with normals $\cos(\theta_i)\hat{a}_x + \sin(\theta_i)\hat{a}_y$ (θ_i is the angle of the i^{th} plane with the y-axis) are intersected with the nose surface. This is shown in Fig. 8-3-a. Then the minima of each curve is computed using,

$$\mathbf{SMIN_m} = V_1(\mathbf{S}_\beta^\mathbf{m}) \tag{8.2}$$

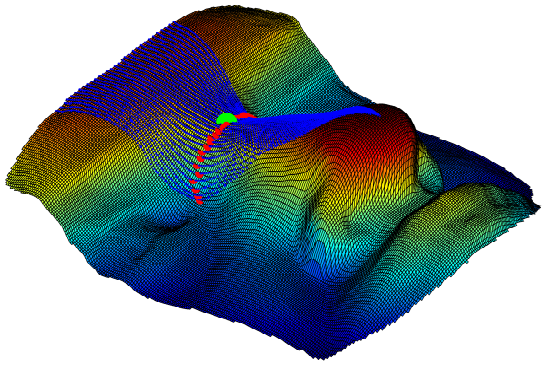
in which, $\mathbf{S}_\beta^\mathbf{m}$ is the m^{th} curve which is found from a β rotated nasal region along the pitch direction. $\mathbf{SMIN_m}$ represents the global minimum per curve. Then, finally the global maximum of the minima from all the curves would correspond to the initial location of the nasal root location. In other words, all the minima computed from the curves ($\mathbf{SMIN_m}$) can be represented as a new curve and therefore, if the $V_1(.)$ function is applied to $-\mathbf{SMIN_m}$, the global maximum would be detected ($\mathbf{L1^0}$), i.e.,

$$\mathbf{L1^0} = V_1(-\mathbf{SMIN_m}). \tag{8.3}$$

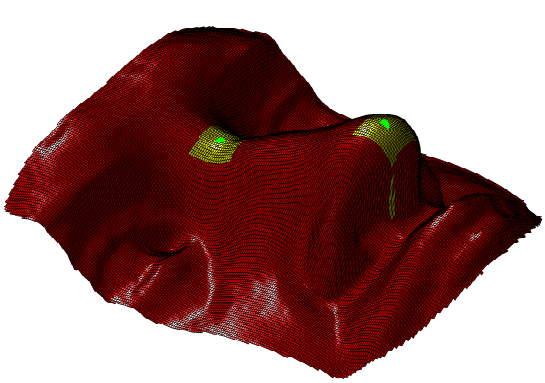
The found nasal root and tip locations ($\mathbf{L1^0}$ and $\mathbf{L4^0}$) might be slightly inaccurate due to the depth variations caused by the noise and expressions. In order to improve their location, for the points situated on a 5×5 (mm^2) area (shown in Fig. 8-3-b) around the nasal root and saddle, the following angular deviation is calculated,

$$\theta_z^i = \arccos \frac{\left|[L4_x^i - L1_x^i, L4_y^i - L1_y^i] \cdot \hat{a}_y\right|}{\left|[L4_x^i - L1_x^i, L4_y^i - L1_y^i]\right|}, \tag{8.4}$$

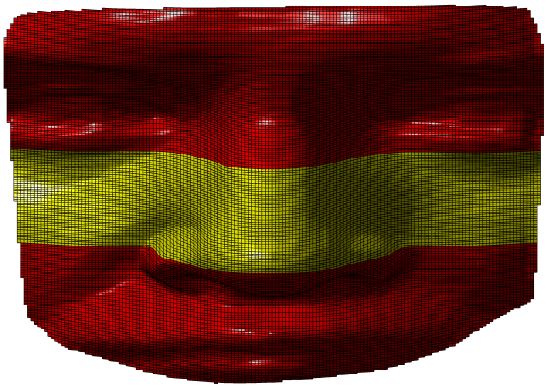
in which, $L1_x^i$, $L1_y^i$, $L4_x^i$, and $L4_y^i$ are found using the current points $\mathbf{L1^i}$ and $\mathbf{L4^i}$ in the area around $\mathbf{L1^0}$ and $\mathbf{L4^0}$, and $\hat{a}_y = [0, 1]$ is the unit vector along the y-axis. θ_z^i is used to rotate the nose region along the roll direction and around the i^{th} nose tip ($\mathbf{L4^i}$). Then the image is divided into a left and right sides and for a strip shown in Fig. 8-3-c, the following objective function is calculated,



(a)



(b)



(c)

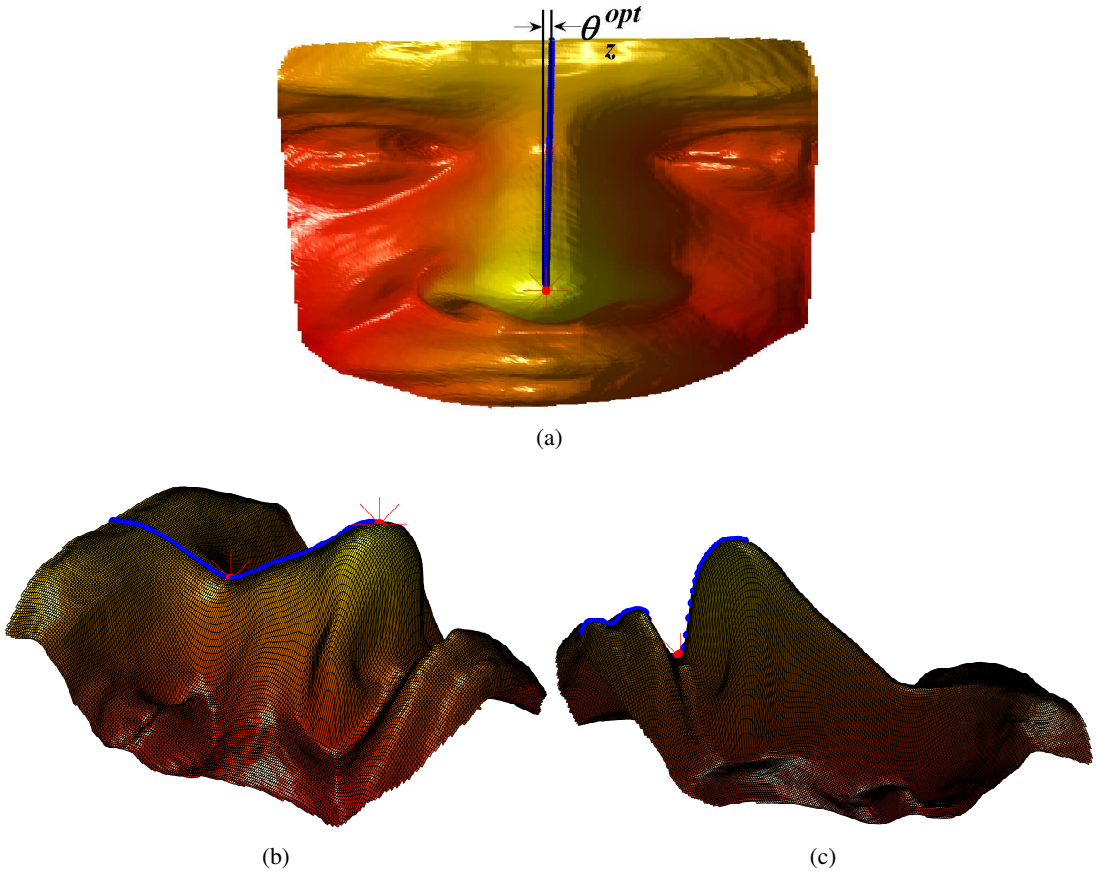
Figure 8-3: (a) The approximate nasal root detection procedure: First curves are intersected with the nasal region (blue points). Then their minima detected using (8.2) (red points). Finally, their maximum is detected as the root's location. (b) RoI for the accurate nose tip and root detection. The green points are the initial approximate locations. (c) Horzontal strip found using the i^{th} nose tip and root locations ($\mathbf{L1^i}$ and $\mathbf{L4^i}$).

$$\begin{cases} \text{find } \theta_z^{opt} \leftarrow \theta_z^i \text{ if} \\ E^i(\mathbf{L1}^i, \mathbf{L4}^i) = \max\left(\sum_y \left[\mathbf{Z}_L^i(x,y) - \mathbf{Z}_R^i(x,y)\right]\right). \end{cases} \tag{8.5}$$

$\mathbf{Z}_L^i(x,y)$ and $\mathbf{Z}_R^i(x,y)$ are the left and right side of the depth map of the i^{th} points in the neighbourhood. Those two points that minimise E^i ($\mathbf{L1}^{opt}$ and $\mathbf{L4}^{opt}$) are the ones whose left and right side depth map has the lowest maximal difference. In other words, they represent the most symmetrical depth map and provide a symmetry plane for the nose depth image. Also, $L1_x^{opt}$ and $L4_x^{opt}$ correspond to the x values of the accurate nose tip and root locations. Therefore, a plane is eventually intersected with the nose surface, along θ_z^{opt} with normal vector $[\cos\theta_z^{opt}, \sin\theta_z^{opt}]$. The locations of the maximum and minimum of the resulting curve are used as $L1_y^{opt}$ and $L4_y^{opt}$. This procedure is depicted in Fig. 8-4. Finally, θ_z^{opt} is used to rotate the

nose region's point clouds around $L4_y^{opt}$ ($L1_y^{opt}$'s location is correspondingly updated). **L4** and **L1** will be $[L4_x^{opt}, L4_y^{opt}]$ and $[L1_x^{opt}, L1_y^{opt}]$, respectively.

The other landmark detected on the nasal region is the subnasale (**L5**). First, a plane with unit vector $[\cos\theta_z^{opt}, \sin\theta_z^{opt}]$, passing the nose tip is intersected with the nose surface. The lower part of the nose tip is cropped and the location of the lowest valleys of the resulting curve (**S**) is chosen as subnasale (**N**), $\mathbf{N} = V_1(\mathbf{S}_{\pi/3})$. This is shown as the red point in Fig. 8-4-c.



(a)

(b) (c)

Figure 8-4: (a) Updating the nasal region using θ_z^{opt}; (b) The maximum and minimum of a curve connecting the optimum $[L4_x^{opt}, L4_y^{opt}]$ and $[L1_x^{opt}, L1_y^{opt}]$ (blue curve) are used as **L4** and **L1**, respectively (red points); (c) Blue points: the cropped nose symmetrical curve; Red point: the lowest minimum, detected as subnasale.

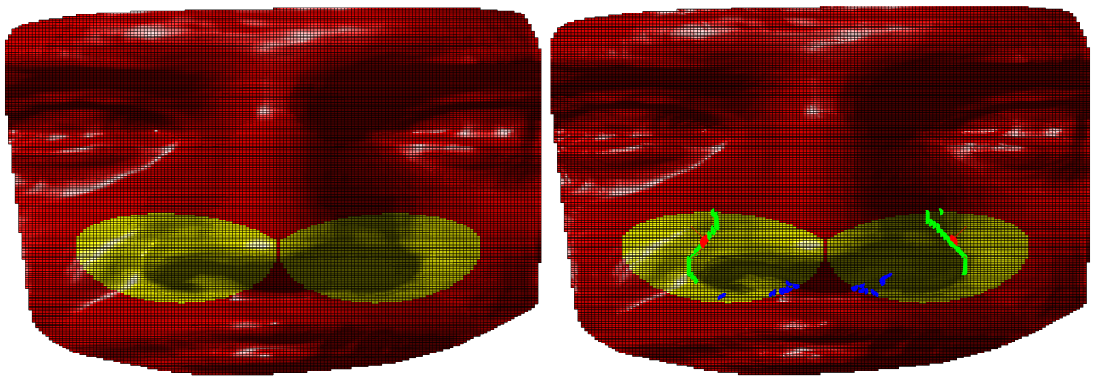
8.2.3 Nose alar groove

The ROI to detect nasal alar is plotted in Fig. 8-5-a. Assuming that the nose tip (**L4**) is located at the origin, the ROI can be found,

$$r = \begin{cases} r_0 \cos^{a_1}(\theta) & 0 \leq \theta < \pi \\ r_0 \cos^{a_2}(\theta) & \pi \leq \theta < 2\pi \end{cases} \tag{8.6}$$

where r_0, a_1 and a_2 are scalar constants determining the length and directivity of the lobes and are chosen to be able to crop the nasal alar region, while avoiding redundant parts (in this work, $r_0 = 30mm$, $a_1 = 4$ and $a_2 = 0.75$). After the ROI is cropped from the nasal region, a plane parallel with the xz-plane is intersected with each row of the ROI. For each intersection, a curve is found and (8.1) is used to find its minima. Three minima are detected per intersection,

$$\begin{cases} \mathbf{RMIN_j} = V_3(\mathbf{Q^j}_{\pi/4}) \\ \mathbf{LMIN_j} = V_3(\mathbf{Q^j}_{-\pi/4}) \end{cases} \tag{8.7}$$



(a) (b)

Figure 8-5: (a) RoI for the nasal alar groove landmarks. (b) Green points (inliers), blue points (outliers) and red points selected locations for **L3** and **L6**.

$\mathbf{RMIN_j}$ and $\mathbf{LMIN_j}$ are 1×3 vectors storing not more than the 3 lowest valleys of the j^{th} row of the ROI, $\mathbf{Q^j}$, for the right and left sides of **L4**, respectively. Then for each row, $\mathbf{RMIN_j}$ and $\mathbf{LMIN_j}$ are compared and the most symmetrical pairs with respect to **L4** are selected. This will also make the detection more robust if by any chance, there was a hole in the intersected curve. After this process, a set of points are found over the nasal alar region, which are depicted in Fig. 8-5-b.

Oulier removal

The points found as candidates for nose alar groove include some outliers as can be seen from Fig. 8-5-b. This is because of noise and deformations on the face due to different expression. To remove the outliers, an iterative approach is used. For a given number of iterations, the

standard deviation (STD) of the 3D Euclidean distance between each point on the j^{th} row ($\mathbf{RMIN_j}$ or $\mathbf{LMIN_j}$) and the next row's point ($\mathbf{RMIN_{j+1}}$ or $\mathbf{LMIN_{j+1}}$) is computed. Then the resulting distance vector's STD is calculated and those points whose STD is higher than a given threshold (in mm) are removed.

The process continues until the number of points remain unchanged. Compared to the outlier removal method explained in the [2], which used K-means clustering to determine the outliers, the method explained here is deterministic and does not have the vulnerability of K-means' empty clustering issue. The outlier removal method is then applied to the points, resulting in the green points between selected as the inliers (Fig. 8-5-b). The left and right pairs, which have the closest y to $\mathbf{L4}$'s are eventually selected as $\mathbf{L3}$ and $\mathbf{L6}$.

8.2.4 Eye corners

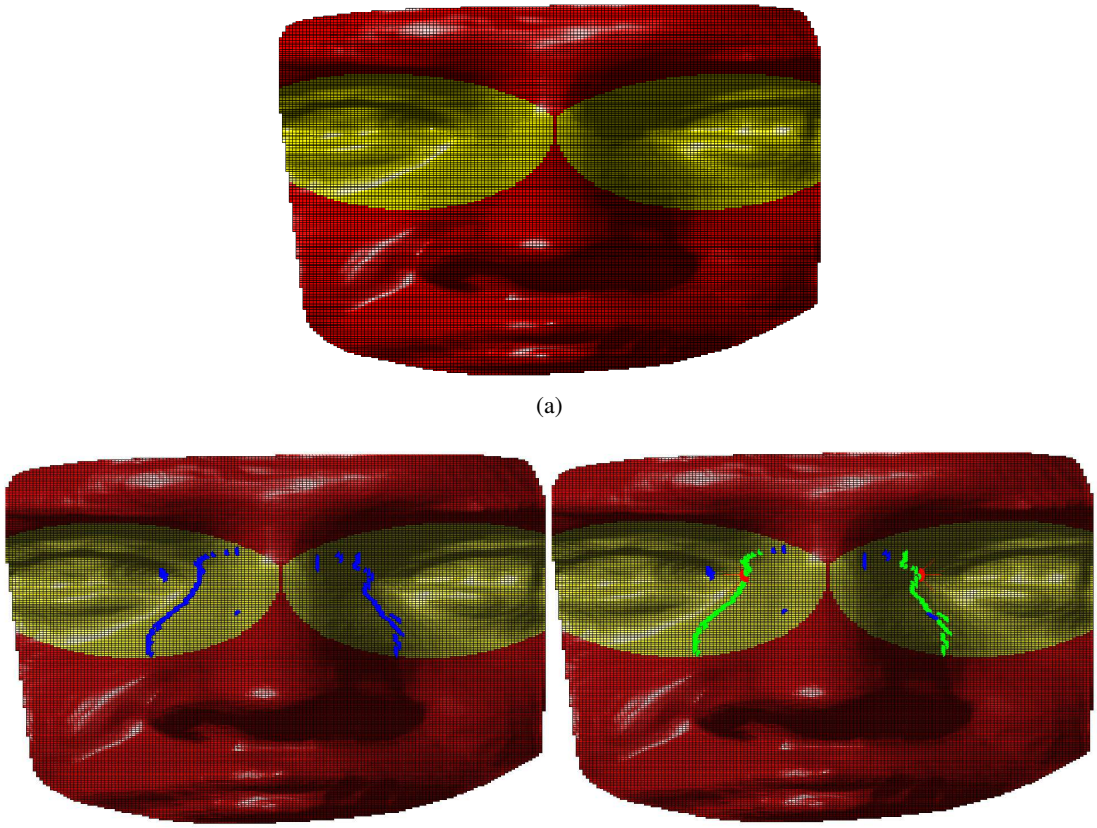
The ROI used to detect the eye corners ($\mathbf{L2}$ and $\mathbf{L7}$) is depicted in Fig. 8-6-a and is based on the following equation while having the nasal root located at the origin ($r_0\prime = 45(mm)$, $a_3 = 4$ and $a_4 = 0.75$),

$$r\prime = \begin{cases} r_0\prime \cos^{a_3}(\theta\prime) & 0 \leq \theta\prime < \pi \\ r_0\prime \cos^{a_4}(\theta\prime) & \pi \leq \theta\prime < 2\pi \end{cases} . \tag{8.8}$$

Similar to the previous section, the region is first cropped and then a plane is intersected with each row of the surface. This will result in a set of curves, whose minima are detected using the valley detector (8-6-b),

$$\begin{cases} \mathbf{RMIN'_k} = V_3(\mathbf{Q'^k}_{\pi/4}) \\ \mathbf{LMIN'_k} = V_3(\mathbf{Q'^k}_{-\pi/4}) \end{cases} \tag{8.9}$$

in which $\mathbf{RMIN'_k}$ and $\mathbf{LMIN'_k}$ are 1×3 vectors including not more than 3 lowest valleys of the k^{th} curve ($\mathbf{Q'^k}$), on the right and left side of the nasal root, respective. For each curve the most symmetrical pairs are selected. All the found points are plotted in Fig. 8-6-b. The outlier removal explained in Section 8.2.3 is applied on the points, resulting in the removal of blue points in Fig. 8-6-c. Finally, the lowest points of $\mathbf{RMIN'_k}$ and $\mathbf{LMIN'_k}$ (the green points in Fig. 8-6-c) are detected and the pair with the most similar distance to the nose tip ($\mathbf{L4}$) are selected as the eye corner landmarks ($\mathbf{L2}$ and $\mathbf{L7}$).



(a)

(b) (c)

Figure 8-6: (a) RoI for eye corners detection; (b) Initial candidates; (c) Inliers denoted in green and eye corners in red (**L2** and **L7**).

8.3 Feature type

The landmarks found in the previous section are the feature points, which are used for the feature descriptors. Before explaining the feature description methodology, in this section the different feature types which have been considered in the work are described. The different features will result in layers of feature maps, in which each layer relates to a different feature type. These layers are then localised by the feature descriptors to create the feature space.

8.3.1 Depth map

Similar to the NCM algorithm, explained in the previous chapter [2], depth is the most straightforward and fundamental feature type which can create the feature space. It does not require any further computation on the point clouds after preprocessing. However, there are some major issues with directly utilising the depth as input for the feature space generator, as described

below.

1. The depth map is sensitive to the noise, as discussed in the previous chapters.
2. If the alignment algorithm fails because of the noise or deformations caused by expressions, depth can not be used as a rotation invariant feature. When the pose correction algorithm fails to correctly align the image, the depth values are changed. Therefore, scatter of the within-class similarity will not be preserved, features may collide with other classes and the recognition performance deteriorates.
3. Depth can not overcome errors caused by the landmarking algorithms. As a consequence, dislocation of landmarks, due to errors from an inconsistent landmarking algorithm, can result in different depth values in the feature space, which is undesirable.

8.3.2 Curvature

One of the most popular approaches to resolving the depth map's sensitivity to pose variation is to extract curvature information for the feature space. Curvature calculation is roll rotation invariant and not sensitive to the point cloud's storage order (i.e. no prior information about the $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ is needed). However, the point clouds are still sensitive to yaw and pitch pose variation, which means a rotated face rotated around the yaw direction would have a different curvature maps from its original pose. Due to the differentiation computation, curvature calculation also amplifies high frequency noise in the data. Moreover, the thresholds assigned to the curvature maps to segment the facial surface are very sensitive to the face's pose and noise. Over-segmentation is usually inevitable and post-processing is essential.

8.3.3 Geodesic distance

A geodesic distance (or simply geodesic) is the shortest curve or line, which can connect two points on a surface. These are evaluated as a feature space descriptor, because of their high simplicity. They are computed by calculating the plane orthogonal to the xy plane, which passes $\mathbf{P_1}$ and $\mathbf{P_2}$, where $\mathbf{P_1}$ and $\mathbf{P_2}$ are given pair of nasal landmarks. The equation for such a plane can be easily found using,

$$\begin{cases} \hat{n} = \frac{(\mathbf{P_1}-\mathbf{P_2})}{||\mathbf{P_1}-\mathbf{P_2}||_2} \times \hat{a}_z \\ \hat{n}_x(x-\mathbf{P_1}_x) + \hat{n}_y(y-\mathbf{P_1}_y) + \hat{n}_z(z-\mathbf{P_1}_z) = 0 \end{cases} \tag{8.10}$$

in which $\hat{n} = [\hat{n}_x, \hat{n}_y, \hat{n}_z]$ is the normal vector for the orthogonal plane. Then the intersection of the plane with the nasal surface is found, which in the discrete space, results in a set of

points. The points are then counted and normalised to approximate the geodesic for between $\mathbf{P_1}$ and $\mathbf{P_2}$.

The pairwise geodesic distance produces a low dimensional feature space. For a set of N_L nasal landmarks, $\frac{(N_L)(N_L-1)}{2}$ distances are found. However, it is again sensitive to the nasal landmarking algorithm's inaccuracy and inconsistency, and noise on the nasal surface. In addition, even for zero noise power and landmarking inaccuracy (which is impossible in reality), there is no biological evidence that the pairwise geodesic distance between the nasal landmarks preserves the within/between class similarity/dissimilarity and so, can be used as a highly discriminant feature. The geodesic distances are also very sensitive to facial expressions. Since each distance is found by approximating the curve length connecting a pair of landmarks, any deformation on the face can deteriorate the curve's geometry and, as a consequence, its length.

8.3.4 Normal vectors

Surface normals have recently been widely used for 3D face recognition [38, 85]. They can provide a rotation invariant feature space. Their superiority in obtaining very localised (pixel-wise) curvature information, unlike the mean and Gaussian curvature no requirement for thresholding, no need for vector length normalisation (as their length is already one) and relatively consistent feature descriptors have been shown by previous researches [85]. Also, the other very important feature of normals is that if the process of depth map creation is slightly inaccurate, due to the alignment or other preprocessing steps, normal vectors can still maintain the curvature information, which can be contributed in the feature space. In other words, normal vectors can add more information to the feature space by integrating the resolution (coordinate) maps (**X** and **Y**) and depth maps (**Z**).

For an input nasal region, represented by its point clouds as $\mathbf{N} = [\mathbf{N_x}, \mathbf{N_y}, \mathbf{N_z}]$ the normals $\mathbf{n}$ are obtained by,

$$\mathbf{n} = [\mathbf{n_x}, \mathbf{n_y}, \mathbf{n_z}] = \nabla \mathbf{N} \tag{8.11}$$

in which $\mathbf{n_x}(i,j) + \mathbf{n_y}(i,j) + \mathbf{n_z}(i,j) = 1$. Despite the normals excellent advantages, they are sensitive to the noise in the data. The gradient of the surface also makes the effects of the high frequency noise more salient. Moreover, using only the normals as the features does not preserve the scale information.

8.3.5 Gabor-wavelets

Let $\mathbf{G_{s,o}}$ be the complex Gabor wavelet at the s^{th} scale and o^{th} orientation levels which, as will be shown later, has been resampled to have the same size as the input image. The wavelet is defined using [158],

$$\mathbf{G_{s,o}} = |\mathbf{G_{s,o}}| \exp(j \arg(\mathbf{G_{s,o}})) \tag{8.12}$$

in which $j = \sqrt{-1}$. The absolute part of $\mathbf{G_{s,o}}$ is found using [158] ($\circ$ is the Hadamard product operator),

$$\begin{cases} |\mathbf{G_{s,o}}| = \frac{a^{(s_m - s)}}{2\pi\sigma_x\sigma_y} \exp(-0.5\{\frac{\mathbf{X_o} \circ \mathbf{X_o}}{\sigma_{xg}^2} + \frac{\mathbf{Y_o} \circ \mathbf{Y_o}}{\sigma_{yg}^2}\}) \\ \arg(\mathbf{G_{s,o}}) = \frac{2\pi\Omega_h}{a^{s_m - s}} \mathbf{X_o} = \frac{2\pi\Omega_h}{a^{s_m - s}} (\cos(\frac{\pi}{o_m(o-1)})\mathbf{X} + \sin(\frac{\pi}{o_m(o-1)})\mathbf{Y}) \\ \sigma_x = \frac{1}{2\pi U_\sigma} \\ \sigma_y = \frac{1}{2\pi V_\sigma} \\ U_\sigma = \frac{(a-1)\Omega_h}{(a+1)a^{s_m - s}\sqrt{2\ln 2}} \\ V_\sigma = \tan(\frac{\pi}{2o_m})(\frac{\Omega_h}{a^{(s_m - s)}} - (2\ln 2)\frac{a^{(s_m - s)}(U_\sigma^2)}{\Omega_h})(\frac{1}{\sqrt{2\ln 2 - \frac{1}{U_\sigma^2}(-2\ln 2 \frac{U_\sigma^2 a^{(s_m - s)}}{\Omega_h})^2}}) \end{cases} \tag{8.13}$$

in which s_m, o_m, σ_x^2 and σ_y^2 are the maximum scale level ($s = \{1, \ldots, s_m\}$), maximum orientation level ($o = \{1, \ldots, o_m\}$), and the filter's horizontal and vertical variances, respectively. a is a scaling factor and is equal to $\sqrt[(s_m - 1)]{\Omega_h / \Omega_l}$, and $\mathbf{X_o}$ and $\mathbf{Y_o}$ are the rotated versions of $\mathbf{X}$ and $\mathbf{Y}$ for the current orientation level (o), which are found after rotating the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps for the input nasal regions around their central point,

$$\begin{bmatrix} \mathbf{X_o} \\ \mathbf{Y_o} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\pi}{o_m(o-1)}) & \sin(\frac{\pi}{o_m(o-1)}) \\ -\sin(\frac{\pi}{o_m(o-1)}) & \cos(\frac{\pi}{o_m(o-1)}) \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \tag{8.14}$$

.

Using this formulation, the size of the output filter $\mathbf{G_{s,o}}$ will be the same as the input image. Therefore, the dot multiplication is quite straightforward. The Fourier transform of the filter is denoted as $\mathcal{F}\{\mathbf{G_{s,o}}\} = \mathbf{G_{s,o}^f}$ and the average of the filter is set to zero ($\mathbf{G_{s,o}^f}(0,0) = 0$) to avoid amplification of the input matrix's average. The filtering is then performed by multiplying the filter's frequency components with Fourier transform of the input depth image ($\mathbf{Z^f}$). Then, the inverse Fourier transform is computed and its absolute value is saved as the filtered image $\mathbf{Z_{G_{so}}}$,

$$\mathbf{Z}_{\mathbf{G_{so}}} = \left| \mathcal{F}^{-1}\{\mathbf{G}^{\mathbf{f}}_{\mathbf{s,o}} \cdot \mathbf{Z}^{\mathbf{f}}\} \right|. \tag{8.15}$$

8.3.6 Gabor-wavelet filtered Normals

In order to integrate the features of the normal maps and Gabor wavelets, in obtaining different curvature features from various scales and orientations, instead of directly applying $\nabla(.)$ to the depth map, it is utilised on the maximum image for all Gabor filters' orientation per scale. In other words, first for each scale, the orientated filters are convolved with the depth map. This is performed in the frequency domain similar to (8.15). The convolution will be multiplication in space domain and therefore, the filters response is multiplied with the image's Fourier transform. Then the absolute value of the inverse Fourier transform is computed. The maximum of all these maps are utilised by $\nabla(.)$.

$\mathbf{G}_{s,o}$ is the Gabor wavelet for the s^{th} and o^{th} scale and orientation level, respectively ($s = \{1, 2, \ldots, s_m\}$ and $o = \{1, 2, \ldots, o_m\}$). The depth map is convolved with the filters. Then for each scale, the maximum of the absolute value of the filtered image is computed. The normal vectors are finally found for the resulting aligned depth map of the nose ($\mathbf{N_z}$) for each scale, i.e,

$$\begin{cases} s = 1, 2, \ldots, s_m \\ \mathbf{n}_s = \nabla \max\{|\mathbf{G}_{s,1} * \mathbf{N_z}|, |\mathbf{G}_{s,2} * \mathbf{N_z}|, \ldots, |\mathbf{G}_{s,o_m} * \mathbf{N_z}|\} \end{cases}. \tag{8.16}$$

where $\mathbf{n}_s$ contains the normal vectors for the s^{th} scale level $\mathbf{n_x}_s$, $\mathbf{n_y}_s$, and $\mathbf{n_z}_s$. Finally feature descriptors (which are explained in the next section) are applied on the nose. Each descriptor correspond to a part of the nasal region, from which the normal vectors of the Gabor wavelets can be extracted. The histogram of each resulting feature vectors, per the x, y and z maps are concatenated to create the feature space. This procedure is demonstrated in Fig. 8-7, for $s_m = 3$ and $o_m = 4$.

8.4 Feature descriptors

The layers produced by Gabor-wavelet analysis for normal vectors calculation would be too large to be used as the feature vectors. It also has too much redundant information and post processing and feature detection is necessary. Moreover, some parts of the nose are essentially more sensitive to the variations caused by expressions. Therefore, further segmentation of the nose can provide the capability to extract the less sensitive parts, while preserving the discriminative regions. For this purpose, three types of feature descriptors are used, which

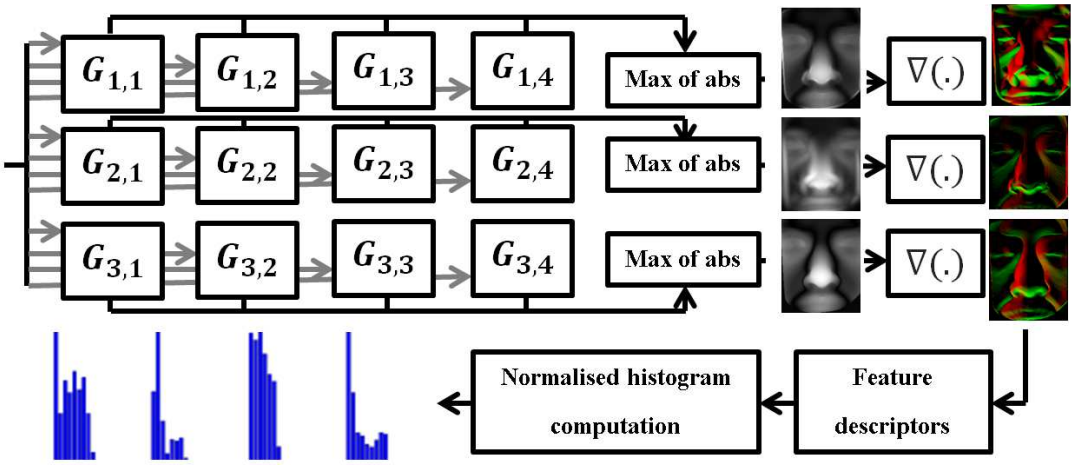


Figure 8-7: The overall feature space creation procedure: 1) the wavelets are applied in different orientation and scales; 2) normals are computed on the maximum of filtered images absolute per scale; 3) feature descriptors are applied; 4) normalised histograms are concatenated for all descriptors.

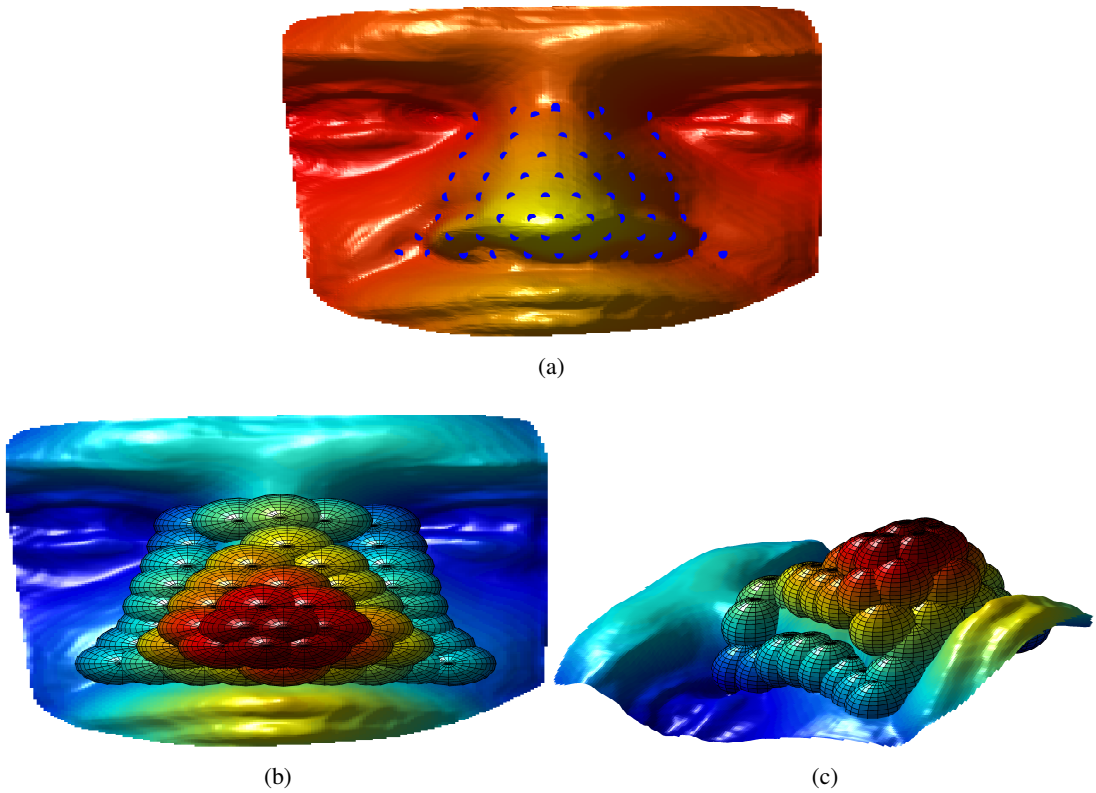
are explained in the following subsections: 1) spherical patches, 2) triangular patches, and 3) curves.

8.4.1 Spherical patches

The basic landmarks ($\{\mathbf{L1}, \mathbf{L2}, \ldots, \mathbf{L7}\}$) are used to create a new combination of points shown in Fig. 8-8-a. The new points are easily obtained by horizontally and vertically dividing the lines, which connect the landmarks. Then for each point, a sphere centralised on the point is intersected with the nasal surface and its inner parts are cropped. Then the histogram of the normal of Gabor-wavelet filtered images are saved as a feature vector. The intersection process is depicted in Fig. 8-8-b, and -c. Various spheres with a given radius (in this case 7 mm) are intersected with the nose. The reason of using this type of feature descriptors is to evaluate the potential of overlapping spherical regions on the nasal surface as feature vectors.

8.4.2 Triangular patches

For this part the combination of points shown in Fig. 8-9-a are used. Selecting sets of three landmarks from Fig. 8-9-a creates a set of triangular patches on the nose. The inner parts of each patch are extracted and used to create the feature space. The main motivation of using the triangular patches is to extract the effects of each triangular part when used for a expression-robust recognition. In order to check the effects of triangles' sizes, two different combination



(a)

(b) (c)

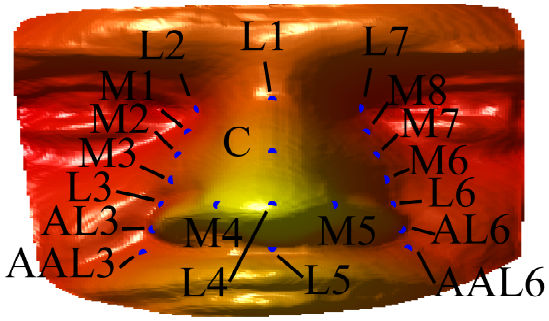
Figure 8-8: (a) Grid of landmarks for the spherical patches creation; (b and c) The spheres centralised on the landmarks and intersected with the nasal surface, which result in the spherical patches on the nose.

of points are used. First the middle landmarks $\{\mathbf{M1}, \mathbf{M2}, \ldots, \mathbf{M8}\}$ are localised as shown in Fig. 8-9-a, by dividing the lines connecting $\mathbf{L2}$ to $\mathbf{L3}$ and $\mathbf{L7}$ to $\mathbf{L6}$ into 4 parts, respectively. Landmark $\mathbf{C}$ is found by dividing the line connecting $\mathbf{L1}$ to $\mathbf{L4}$. And four more auxiliary landmarks ($\mathbf{AL3}$, $\mathbf{AAL3}$, $\mathbf{AL6}$ and $\mathbf{AAL6}$) are localised by extending the lines connecting $\mathbf{L2}$ to $\mathbf{L3}$ and $\mathbf{L7}$ to $\mathbf{L6}$, respectively. The following sets of patches are then found over the nasal surface, which are shown in Fig. 8-9-b and -c,

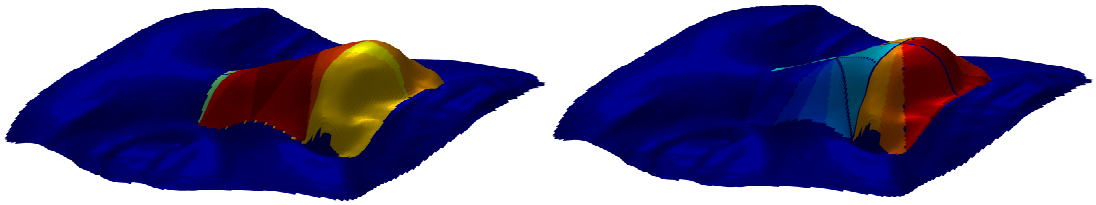
1. Larger triangles (Fig. 8-9-b): {**L3L1L2**, **L6L1L7**, **L3L4L1**, **L6L4L1**, **AAL3L4L3**, **AAL6L4L6**, **L3CL2**, **L6CL7**, **L4L2L7**, **L4L2L1**, **L4L7L1**, **L5L2L7**, **L5L3L6**, **L5L2AAL3**, **L5L7AAL6**, **CL2L7**, **CL3L6**, **L3L7L2**, **L6L2L7**, **L3M7L2**, **L6M2L7**}

2. Smaller triangles (Fig. 8-9-c): {**L1CL2**, **L2CM1**, **M1CM2**, **M2CM3**, **M3CL3**, **L3CM4**, **M4CL4**, **L1CL7**, **L7CM8**, **M8CM7**, **M7CM6**, **M6CL6**, **L6CM5**, **M5CL4**, **L4L3AL3**, **L4AL3AAL3**, **L4L6AL6**, **L4AL6AAL6**, **L4L5AAL3**, **L4L5AAL6**, **AAL6AAL3L5**}

Similar to the spherical patches, the histogram of Gabor-wavelet normals for each triangle's inner part is stored as a feature vector.



(a)



(b) (c)

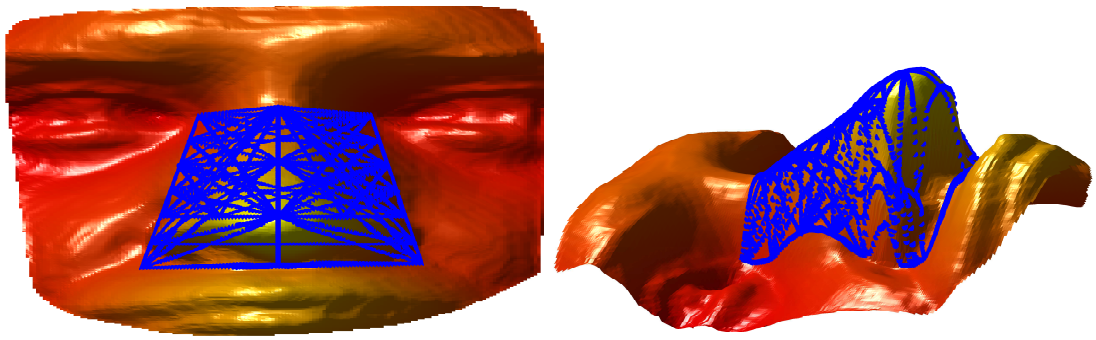
Figure 8-9: (a) The combination of landmarks used to create the triangular patches; (b) Larger triangular patches; (c) Smaller triangular patches.

8.4.3 Curves

In this work, using different pairs of landmarks, an orthogonal plane can be found. The plane is then intersected with the nose surface to find the curves. For instance, assuming $\mathbf{A_1}$ and $\mathbf{A_2}$ as two landmarks selected from the ones shown in Fig. 8-9-a, the plane's normal that connects the points and is orthogonal to the xy plane will be,

$$\hat{p}_{\mathbf{A_1A_2}} = \hat{a}_z \times \frac{[\mathbf{A_1} - \mathbf{A_2}]}{\sqrt{[\mathbf{A_1} - \mathbf{A_2}]}}. \tag{8.17}$$

The curve on the nose are used to extract features. In other words for each curve, the histogram of the normal vectors of the Gabor wavelet filtered nose images are used as the feature vector. The set of landmarks used for creating the curves are as followed: {**L1L2**, **L1M1**, **L1M2**, **L1M3**, **L1L3**, **L1AL3**, **L1AAL3**, **L1C**, **L1M4**, **L1L4**, **L1L5**, **L1L7**, **L1M8**, **L1M7**, **L1M6**, **L1L6**, **L1AL6**, **L1AAL6**,**L1M5**, **L2L7**, **L2C**, **L2L4**, **L2L3**, **L2AAL3**, **L2M7**, **L2M8**, **L7C**, **L7L4**, **L7L6**, **L7AAL6**, **L7M2**, **L7M1**, **M1C**, **M2C**, **M3C**, **L3C**, **AL3C**, **AAL3C**, **L4C**, **M8C**, **M7C**, **M6C**, **L6C**, **AL6C**, **AAL6C**, **L4M1**, **L4M2**, **L4M3**, **L4L3**, **L4AL3**, **L4AAL3**, **L4L5**, **L4AAL6**, **L4AL6**,**L4M6**, **L4M7**, **L4M8**, **M4M2**, **M4M3**, **M4M7**, **M5M7**, **M5M6**, **M5M2**, **M1M8**, **M2M7**, **M3M6**, **L3L6**, **AL3AL6**, **AAL3AAL6**, **AAL3L5**, **AAL6L5**, **M2L6**, **M7L3**}. The curves are plotted in Fig. 8-10-a and -b.



(a) (b)

Figure 8-10: The nasal curves: (a) frontal and (b) side views.

8.5 Feature selection and matching using genetic algorithms

In this step, those subsets of the feature vectors, which correspond to the curves and spherical patches that are more vulnerable to expressions are detected. For a given type of feature descriptor, for n different scales $\{s_1, s_2, \ldots, s_n\}$ of Gabor wavelets, the following feature vector is obtained,

$$\begin{cases} \mathbf{F} = [\mathbf{F}_{s_1}, \mathbf{F}_{s_2}, \ldots, \mathbf{F}_{s_n}], \\ \mathbf{F}_{s_k} = \left[\mathbf{Fx}_{s_k}, \mathbf{Fy}_{s_k}, \mathbf{Fz}_{s_k}\right] \end{cases} . \tag{8.18}$$

$\mathbf{Fx}_{s_k}$, $\mathbf{Fy}_{s_k}$ and $\mathbf{Fz}_{s_k}$ are the features of the s_k^{th} scale, for the x, y and z components of the surface normal, respectively. Prior to explaining the feature selector, it is necessary to first formulate the feature space. For K feature descriptors, each feature set of the normal maps is represented by the concatenation of K different histograms computed from the feature descriptors, each of them having the length of the histograms (h_l),

$$\begin{cases} \mathbf{Fx}_{s_k} = [\mathbf{Hx}_{1,s_k}, \mathbf{Hx}_{2,s_k}, \ldots, \mathbf{Hx}_{K,s_k}], \\ \mathbf{Fy}_{s_k} = \left[\mathbf{Hy}_{1,s_k}, \mathbf{Hy}_{2,s_k}, \ldots, \mathbf{Hy}_{K,s_k}\right], \\ \mathbf{Fz}_{s_k} = [\mathbf{Hz}_{1,s_k}, \mathbf{Hz}_{2,s_k}, \ldots, \mathbf{Hz}_{K,s_k}] \end{cases} . \tag{8.19}$$

$\mathbf{Hx}_{i,s_k}$, $\mathbf{Hy}_{i,s_k}$ and $\mathbf{Hz}_{i,s_k}$ are the normalised histograms computed using the i^{th} feature descriptor ($i = 1, \ldots, K$) for the s_k^{th} scale ($k = 1, \ldots, n$) on the normal map $\mathbf{n}_{s_k}$.

Now that the feature vector is formulated, the aim is to find a binary vector to be used as a switch to select or omit feature descriptors. Using a $1 \times K$ binary vector $\mathbf{Bn}$, the following vector $\mathbf{B}_{s_k}$ can be computed for the s_k^{th} scale, which has the same number of columns as of $\mathbf{Fx}_{s_k}$, $\mathbf{Fy}_{s_k}$ or $\mathbf{Fz}_{s_k}$,

$$\begin{cases} \mathbf{B}_{s_k} = [\mathbf{B}_{1,s_k}, \mathbf{B}_{2,s_k}, \ldots, \mathbf{B}_{K,s_k}], \\ \mathbf{B}_{i,s_k} = \begin{cases} \{0,0,\ldots,0\} \text{ if } \mathbf{Bn}(d) = 0 \\ \{1,1,\ldots,1\} \text{ if } \mathbf{Bn}(d) = 1 \end{cases} \end{cases} \tag{8.20}$$

$\mathbf{B}_{i,s_k}$'s elements ($i = 1, \ldots, K$) are all zero or all one, depending on the value of the i^{th} element of $\mathbf{Bn}$. Finally, $\mathbf{B}_{s_k}$ is concatenated for all scales to create,

$$\mathbf{B} = \left[\underbrace{\overbrace{[\mathbf{B}_{s_k}, \mathbf{B}_{s_k}, \mathbf{B}_{s_k}]}^{\text{for all normals in scale 1.}}, \overbrace{[\mathbf{B}_{s_k}, \mathbf{B}_{s_k}, \mathbf{B}_{s_k}]}^{\text{for all normals in scale 2.}}, \ldots, \overbrace{[\mathbf{B}_{s_k}, \mathbf{B}_{s_k}, \mathbf{B}_{s_k}]}^{\text{for all normals in scale n}}}_{\text{length}=s_n \times 3 \times K \times h_l} \right]. \tag{8.21}$$

Now if the nucleus binary vector $\mathbf{Bn}$ is changed, the resulting $\mathbf{B}$ would vary. A curve or patch is selected or omitted based on the value of $\mathbf{Bn}$'s elements. If $\mathbf{Bn}(d) = 1$, then the d^{th} curve or patch is selected, otherwise it is omitted. By grouping the neutral samples for the gallery and the non-neutral ones for the test phase, and varying $\mathbf{Bn}$ the most expression robust curves and patches can be selected . The Kernel Fisher's analysis (KFA) algorithm with polynomial kernel is utilised on the feature space to project the features to a lower dimensional space using a supervised approach. The resulting low dimensional samples are matched with those in the gallery using the Mahalanobis cosine distance,

$$\mathbf{D} = -\left(\frac{\mathbf{X_g}}{\sqrt{|\mathbf{X_g} \Sigma^{-1} \mathbf{X_g}^{\mathsf{T}}|}} \right) \Sigma^{-1} \left(\frac{\mathbf{X_t}}{\sqrt{|\mathbf{X_t} \Sigma^{-1} \mathbf{X_t}^{\mathsf{T}}|}} \right)^{\mathsf{T}}. \tag{8.22}$$

Assuming S_g gallery, S_p test and d_p as dimension of the subspace feature space (in all of the experiments, it is equal to S_g), $\mathbf{X_g}$ and $\mathbf{X_t}$ will be $S_g \times d$ and $S_p \times d$ train and test matrices, respectively. Σ is the $d_p \times d_p$ covariance matrix computed over $\mathbf{X_g}$ and $\mathbf{D}$ is a $S_g \times S_p$ distance matrix including the matching scores. It is aimed to maximise the probability of assigning the test samples $\mathbf{X_t}$ to their corresponding classes (subjects), by matching them to gallery samples $\mathbf{X_g}$, while $\mathbf{Bn}$ is varied, i.e

$$\mathbf{Bn}_{opt} = \arg\max_{\mathbf{Bn}} \{R_1\} \tag{8.23}$$

in which R_1 is the average probably that the label corresponding to the smallest matching score found by (8.22), is the same as the test sample's label. In other words, R_1 is the rank one recognition rate, which is maximised by changing $\mathbf{Bn}$. Because of GA's capability in finding the optimal point in high-dimensional binary population type, it is utilised for the optimastion

problem.

8.6 Experimental results

Two dataset are used to evaluate the proposed algorithm. The first of these is the FRGC dataset. The samples in the Spring 2003 folders are known as the v1.0, while the collection in the other two folders constitute v2.0. FRGC v1.0 and v2.0 have 267 subjects (838 samples) and 466 subjects (4007 samples), respectively. In order to evaluate the algorithm on this dataset, three sets of experiments are defined. For the first set, FRGC v2.0 is divided into 466 samples for gallery and 3541 samples for the test. This arrangement has been extensively used in the literature ([97, 85, 32, 70, 99, 36, 108, 159]). The second experiment is known is FRGC's ROC III [14] on Exp III. This is a verification scenario, which uses the between seasons 3D range data. For this experiment, usually EER or 0.1% FAR is reported.

The third evaluation on FRGC is named as expression vs. expression. The dataset consists of samples with neutral and non-neutral expressions. Using different sets of samples with different expression types for the probe set, results in two evaluations: neutral gallery vs. neutral probe and neutral gallery vs. non-neutral probe. The purpose of this experiment is to check the algorithm's robustness against unseen expressions in the gallery.

The second dataset used in this work is the Bosporus 3D face dataset. For this dataset two methods are used for evaluation. The first is to use one neutral sample per subject as the gallery (105 samples) and the rest for the test phase (2797). The second approach is based on using different expressions for gallery and other expressions as probe.

8.6.1 Landmarking consistency and accuracy

The nasal landmark's localisation is the basic of the proposed algorithm. Although its consistency will also be verified from the recognition algorithm's performance, in order to evaluate the landmarking algorithm's consistency, for each individual landmark, first the nose tip is moved to the origin and the rest of the landmarks are correspondingly translated. Then the pair-wise Euclidean distance of each landmark's position per expression for each subject is computed and the average and standard deviation calculated. For a perfect landmarking method, the average distance should be close to zero. However, due to the noise in the data and image acquisition errors, it will deviate from zero. The reason of individually considering each expression is that the landmark's location changes slightly as the facial expression varies. However, the landmark's location should remain reasonably constant for the same expression type. It should be mentioned that although FRGC includes 557 subjects overall, some subjects

have only one sample; these subjects are omitted for the consistency error evaluation, since the pair-wise distance can not be computed. Table 8.1 shows the results on the FRGC and Bosphorus datasets.

Landmarks	**L1**	**L2**	**L3**	**L5**	**L6**	**L7**
Bosphorus	1.06 ± 0.58	1.76 ± 1.03	1.06 ± 0.62	1.11 ± 0.38	1.19 ± 0.60	2.12 ± 1.14
FRGC	2.04 ± 1.09	2.95 ± 1.61	1.29 ± 0.82	1.86 ± 0.85	1.22 ± 0.62	2.91 ± 1.53

Table 8.1: Landmarking consistency error in mm, used to evaluate the within-class similarity of the distribution of the landmarks.

Since the location of the rest of the landmarks are computed using the basic landmarks $\{\mathbf{L1}, \mathbf{L2}, \ldots, \mathbf{L7}\}$, only these landmarks are used for the consistency error evaluation. The error is higher for the FRGC dataset as it includes noisier samples, especially in the Spring 2003 folder. The most consistently accurate pair of landmarks on both datasets are the nasal alar (**L3** and **L6**). The error is slightly higher for the eye corners (**L2** and **L7**). The subnasale (**L5**) location is more consistently detected for the Bosphorus dataset's samples. This is again mainly because of the noise, since for the FRGC samples, the imaging modality is less accurate in reconstructing the regions with higher frequency components, in which there is significant curvature fluctuations.

The other important factor for a landmarking algorithm is its accuracy. In order to evaluate this factor, the landmarks' locations are compared with those manually located by the Bosphorus dataset's providers, which are used as the ground truth. To do this, the same procedure, including the PCA alignment matrix and θ_z are applied to the ground truth landmarks to remap the points to the aligned faces domain. The precision curve ([151]) is then computed for the basic landmarks $\{\mathbf{L2}, \mathbf{L3}, \mathbf{L4}, \mathbf{L6}, \mathbf{L7}\}$. Since the location of nasal root ($\mathbf{L1}$) has not been assigned by the dataset providers, it is excluded from the accuracy computation. The precision curves shown in Fig. 8-11 were found over the action units samples (2150 observations) and samples with neutral and non-neutral expression in the Bosphorus dataset (653 observations), constituting 2803 samples.

The comparison of the landmarking results with those reported in [69] is shown in Table. 8.2. Although the results in [69] were for the expression types, considering the number of samples and the recognition performance, they can be easily merged and computed for all three types of samples (action units, neutral and non-neutral samples).

For the nasal alar and tip, the proposed algorithm has higher accuracy. Especially for the nose tip the accuracy is about 2% higher. However, for the eye corners, the [69]'s method has an approximately 2% better performance, Also, unlike [69], the method does not require a training step. However, [69] is significantly more robust in the cases of occlusion and large

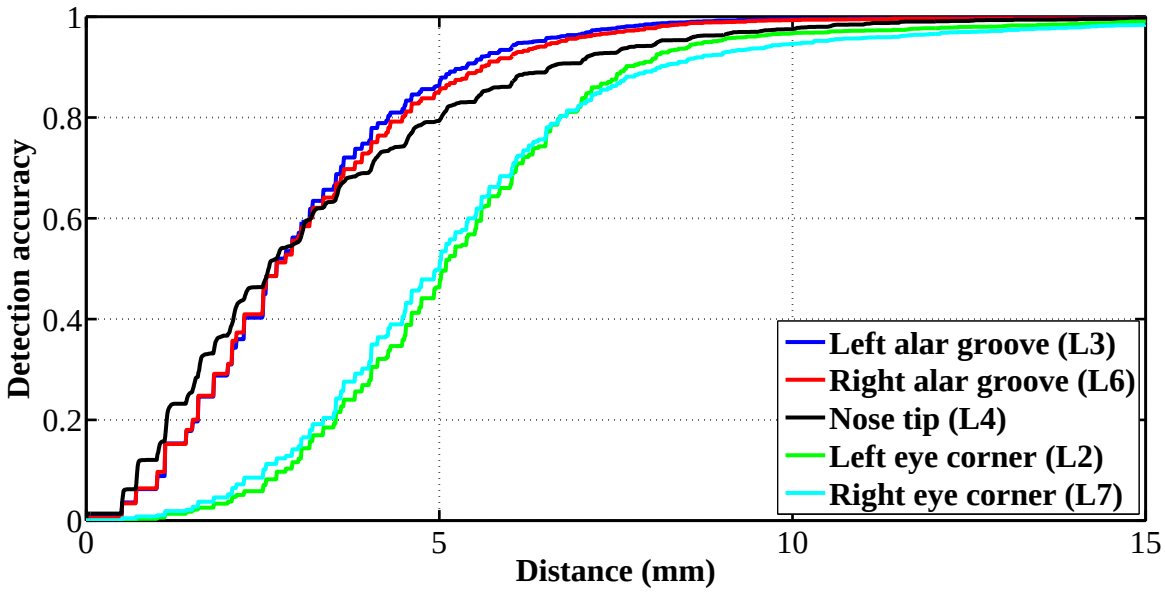


Figure 8-11: Precision curves for the proposed landmarking algorithm on the Bosphorus dataset, used for quantitative evaluation of the landmarking algorithm.

rotations.

8.6.2 Feature space parameters

Spheres radius effect for the circular patches

The circular patches are found, when the grid of landmarks are localised and the radius of each circle is assigned. Choosing small sphere would result in missing the regional information, while too large sphere can deteriorate the expression invariance by adding more expression sensitive regions. Also, a too dense grid produces high redundancy in the feature space and a too sparse distribution might fail to grab all the necessary discriminant features.

In order to evaluate the effects of varying these parameters on the recognition ranks, in this section, the leave-one-out procedure is used for various sets of parameters on the Bosphorus dataset (for its 2797 samples). The result is shown in Fig. 8-12, while 21 bins are used for the histograms computation. The figure verifies the initial hypothesis that there should be an optimal radius for the spheres. When the radius of the spheres is increased, the recognition ranks monotonically increases until the $r = 11mm$. After this point, the recognition performance deteriorates, due to the high redundancy added to the feature space. Moreover, the larger sphere crop regions with higher vulnerability to the variations caused by expression.

The formation of landmarks in Fig. 8-13-a, -b and -c, can be changed by assigning

Algorithm	Threshold in mm	Left alar groove L3	Left alar groove L6	Left eye corner L2	Right eye corner L7	Nose tip L4
Proposed method	< 10	99.55%	99.35%	96.69%	94.59%	97.52%
	< 12	99.62%	99.62%	97.73%	96.56%	99.04%
	< 15	99.66%	99.66%	99.04%	98.38%	99.66%
	< 20	99.69%	99.66%	99.59%	99.62%	99.79%
Creusot *et al.* [69]	< 10	97.96%	98.43%	98.82%	98.50%	95.47%
	< 12	99.18%	99.71%	99.65%	99.43%	98.15%
	< 15	99.82%	99.86%	99.93%	99.75%	98.97%
	< 20	99.90%	99.90%	99.93%	99.86%	99.33%

Table 8.2: Percentage of points within the thresholded distance from the ground truth.

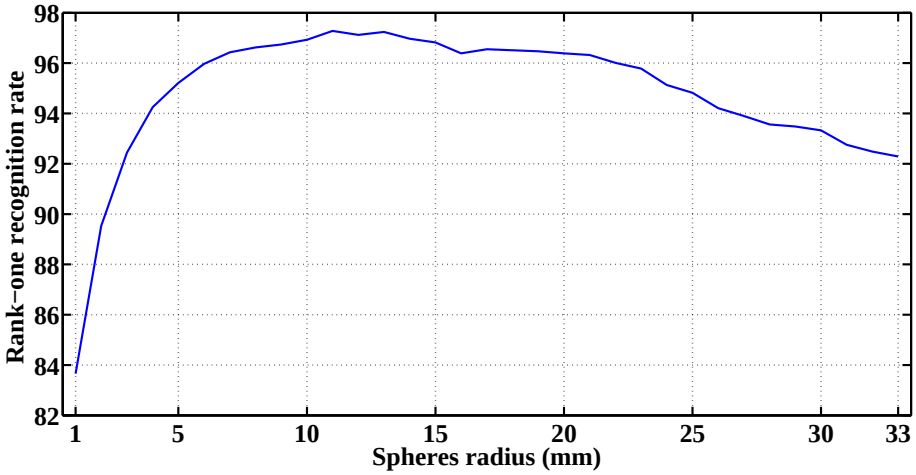


Figure 8-12: Rank-one recognition rate for different radii for spherical patches.

different horizontal and vertical division steps on the lines connecting the basic landmarks $\{\mathbf{L1}, \mathbf{L2}, \ldots, \mathbf{L7}\}$. If d is the number of division for the first horizontal line and m is the number of points in the middle of the two landmarks. For the other pairs of landmarks, the number of middle points linearly increases, resulting in the set of points shown in Fig. 8-13. For instance, when $m = 6$ means that the distance between **L2** to **L3** and **L6** to **L7** are divided into 6 parts, resulting in 5 new points. Then for $d = 3$, the distance between the first pair of horizontal landmarks (the eye corners) are divided into three parts, resulting in two middle landmarks. This procedure repeats for the next horizontal pair of landmarks (Fig. 8-13-c), while the number of middle points linearly increases (for the h_p^{th} horizontal pair, $d + h_p$ middle points are located). Also, when $d \leq 1$, no middle landmark is localised (Fig. 8-13-a).

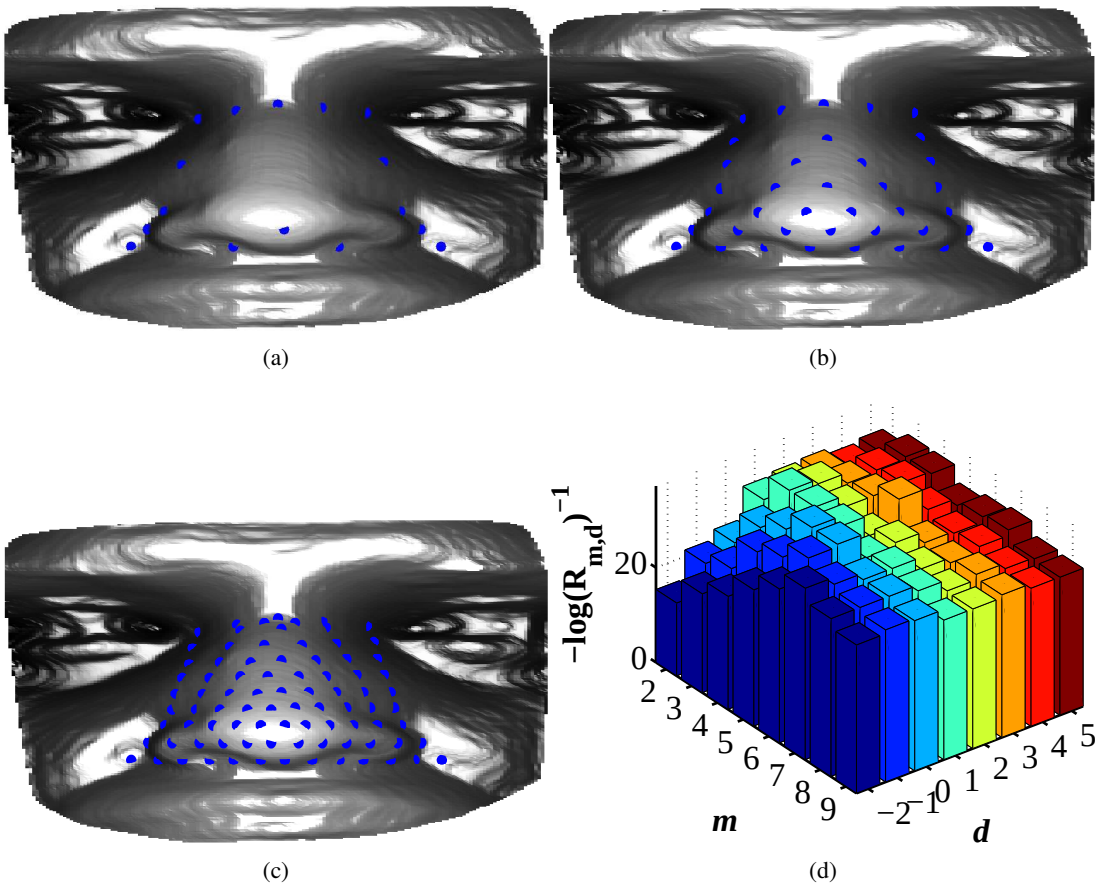


Figure 8-13: Formation of the landmarks for the spherical patches: (a) $m = 2$ and $d = -2$; (b) $m = 4$ and $d = 0$; (c) $m = 6$ and $d = 3$; (d) Recognition rate for different m and d.

The result of applying different combination of landmarks are plotted in Fig. 8-13-b. For this experiment, the neutral samples in the Bosphorus dataset are used for the gallery and the non-neutral samples (including all the samples with different expressions and action units) for the test. $R = 11mm$ is used for the spheres and 21 bins for histograms. In order to make the differences between the recognition ranks more obvious, instead of directly plotting the ranks, their negative inverse logarithm is depicted for each m and d.

As it can bee seen from Fig. 8-13-d, the recognition rank for a given m and d ($\mathbf{R}_{m,d}$) increases for a more dense grid of landmarks. It reaches a maximum when $m = 5$ and $d = 0$ ($\mathbf{R}_{m,d} = 0.9735$, then it decreases for higher values of m and d. It eventually remains constant for $m > 7$ and $d > 3$. The figure shows that a too dense formation of the grid of landmarks does not necessarily increase the recognition performance. Also, another interesting fact from Fig. 8-13-d is that the recognition performance is less sensitive to the horizontal divisions within the nasal region than its boundary. In other words, the surrounding parts of the nose has

more discriminant power than its within and less sensitive to expressions.

Number of histogram bins for triangular patches and curves

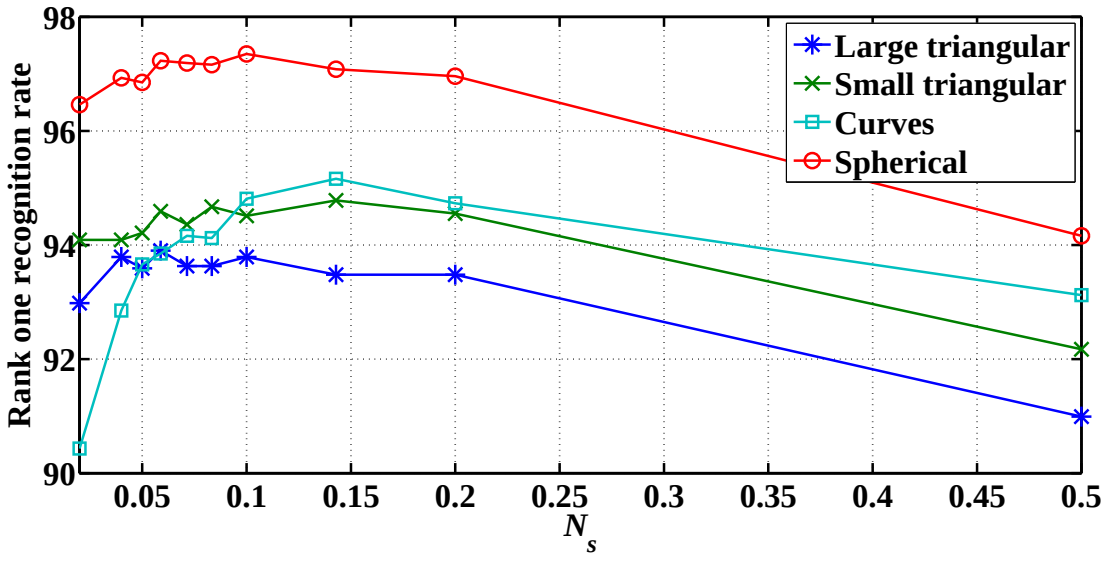
The proposed feature space is based on the histogram of Gabor-wavelet filtered normal vectors. Depending on the number of histogram bins, different sizes of feature vectors are produced. In this part, the effects of varying the histogram bins on the recognition performance is quantitatively evaluated. The results are depicted in Fig. 8-14-a. When the number of bins are increased, the normal maps values can be mapped to a wider range. A big set of bins can spread and quantise the normals to a wider range of numbers, which might produce redundancy in the feature space. Also, another consequence of a large collection of histogram bins is the increase of feature space dimensionality. Since there might not be enough number of training samples for newer dimensions, the recognition performance can reduce (curse of dimensionality). On the other hand, too few bins will concentrate the normal vectors to a tighter range, resulting in a too dense histogram, which can not properly represent the information included in patches or curves.

Let us assume that N_s is the step size number, used to create the histogram bins (for instance, $N_s = 0.5$, results in the following set of histograms bins for the normal vectors' range [-1, 1]: $\{$-1, -0.5, 0, +0.5, +1$\}$). As N_s increases the length of the histograms decreases, and as shown in Fig. 8-14-a, resulting in lower recognition ranks. However, too small N_s does not necessarily increase the performance. It creates a higher dimensional feature space and maps the data to a wider range. The lack of training samples for the new feature space dimensions deteriorates the recognition ranks.

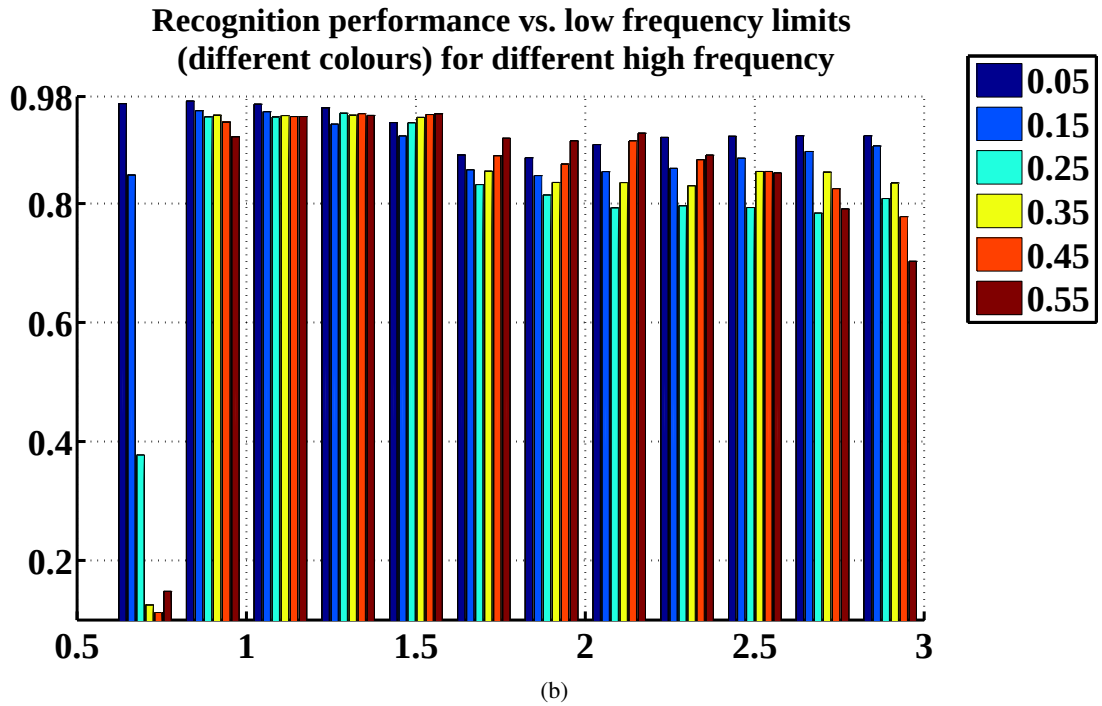
Nasal region's frequency analysis and Gabor-wavelets scale vs. orientation levels

The number of scales used for applying wavelets on the nasal region has direct effects on the smoothing level and feature space dimensionality. Increasing the number of filters' orientations, on the other hand, can highlight edges direction in a better way. However, too large set of orientation and scale levels can increase the dimensionality, images' blurriness and wrong edges. The low and high frequency of filters should be correctly chosen for the Gabor-wavelets filters as well. The result of applying different lower and higher frequencies on the circular patches, with the same configuration as the previous figure 8-14-b (neutral gallery vs. non-neutral probe on Bosphorus), is depicted in Fig. 8-14-a.

Each colour represents a lower frequency level, starting from 0.05 to 0.55. Increasing the Ω_h/Ω_l ratio (or a) increases the variance of the filters in the frequency domain. As a consequence, a wider range of the input range image's frequency components are passed through



(a)



(b)

Figure 8-14: For the spherical patches: (a) Rank-one recognition rate for different histogram bins increment steps. (b) Rank-one recognition performance for different Ω_l (different colours) and Ω_h (x-axis).

the filtering process and the outcome will be highly smoothed image. The discrimination between the faces will be degraded and the recognition rates deteriorates. This can be seen from the higher values of Ω_h in Fig. 8-14-b. On the other hand, lower values for Ω_h can generate

sharper wavelets, which are capable to filter more localised frequency components. The output will also be significantly less blurry and within-class scatter will be more concentrated for each subject. This is verified by lower values of Ω_h in Fig. 8-14-b, where the peak occurs when $\Omega_h = 0.7$ and $\Omega_l = 0.05$.

Figure. 8-15 shows the recognition performance for different maximum scales and orientations (s_m and o_m), when $\Omega_l = 0.05$ and $\Omega_h = 0.05$. The value of s_m has direct effect on the feature space dimensionality (because for each scale, the maximum of all filtered images' absolute values in different orientations are computed). For all the maximum orientation levels, for $s_m \geq 4$ the recognition ranks decreases. This is due to the curse of dimensionality and also producing highly blurry images. On the other hand, for $o_m > 2$, increasing the number of orientations has not significantly affected the recognition performance and for $o_m = 4$ and $s_m = 4$ the highest rank-one recognition rate is obtained at 97.35%. For $s_m = 2$ the lowest rank-one recognition rates are found, which shows that the discriminatory information on the nasal region can not be achieved in low scales. This quantitative result verifies the initial justification to use wavelet's multi-resolution decomposition approach for normal-depth maps instead of utilising a single resolution.

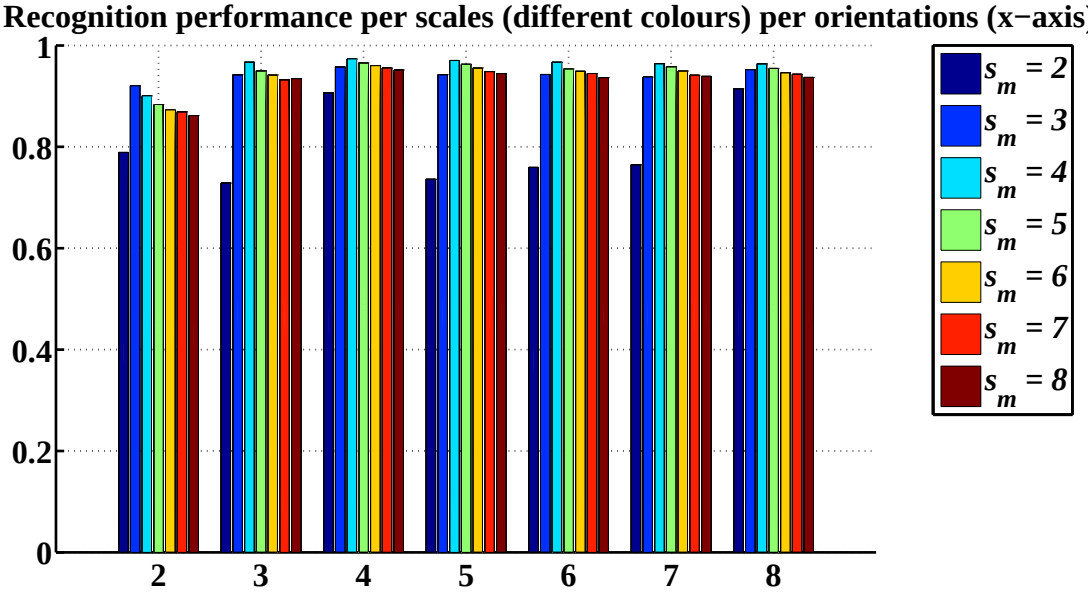


Figure 8-15: Rank-one recognition performance for different maximum scales (s_m) and orientations o_m for Gabor-wavelets applied to the spherical patches.

8.6.3 Expression-robust 3D nose recognition

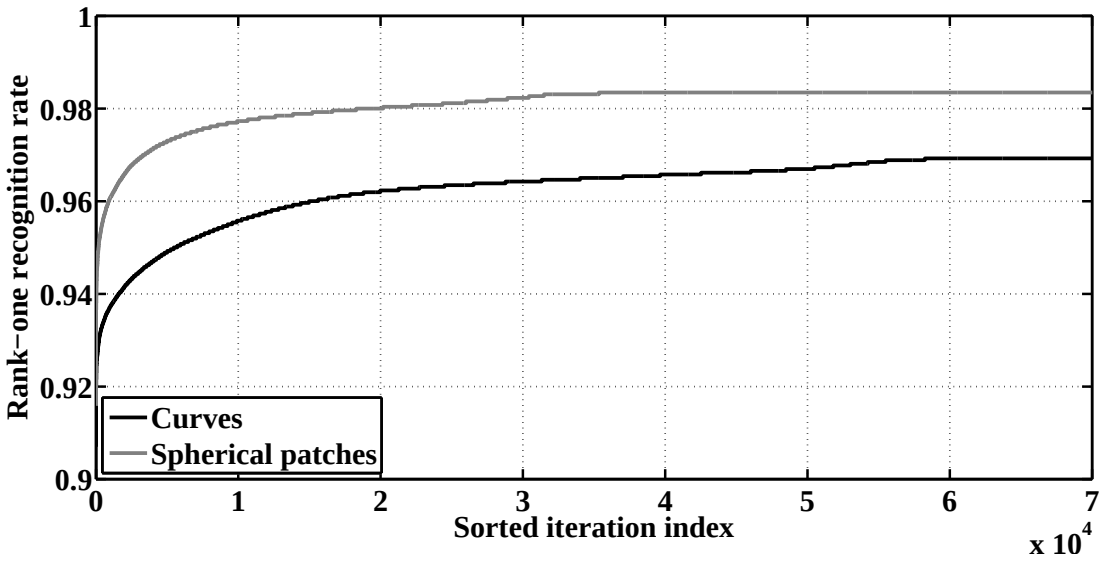
Feature selection

Using the algorithm explained in section 8.5, the most discriminative patches and curves are selected. The feature selection is applied on the Bosphorus dataset, which includes higher variations of expression. The neutral samples are used for training and the non-neutral samples (including anger, disgust, fear, happiness, sadness, and happiness) for the test phase. The KFA with a polynomial kernel is used for subspace projection. Then the Mahalanobis cosine distance in (8.22) is applied for the matching step at every GA's iteration. Since GA is an stochastic optimisation algorithm and the outcome at every iteration is not necessarily in descending order, the ranks are sorted in ascending order and plotted for their corresponding indexes in Fig. 8-16. Four types of feature sets are optimised: spherical, small and large triangular patches, and the nasal curves. The ranks converge at 98.35%, 96.93%, 95.55% and 95.50% for the spherical, small and large triangular and curves, respectively.

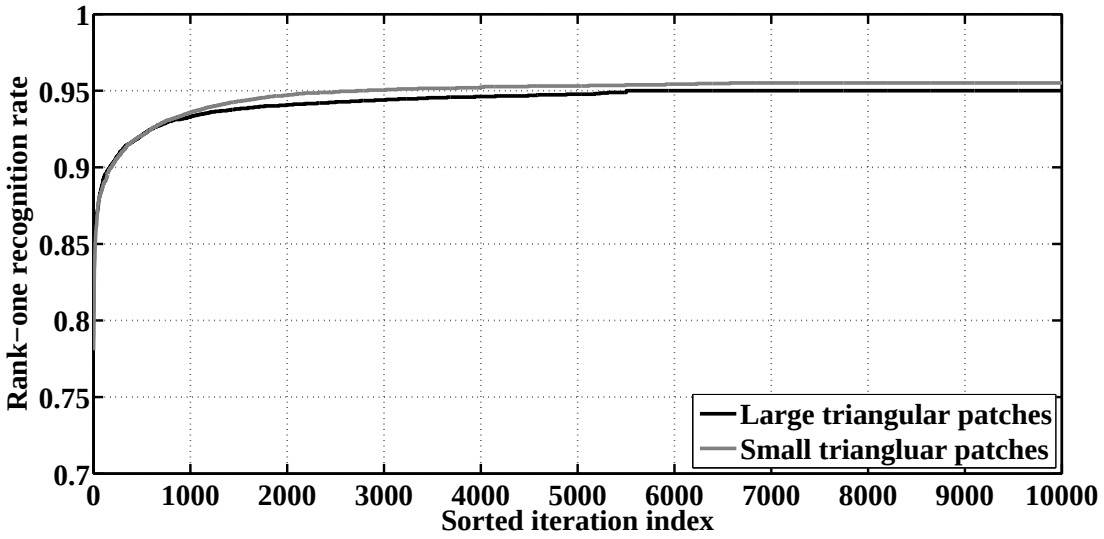
The results for each feature space modality show that which patches or curves are more robust against variations caused by expression. In other words, the most stable regions, which maintain the within-class similarity and between-class dissimilarity are detected from the curves and patches. These dominant patches and curves are plotted in Fig. 8-17 and for the curves and triangular patches are as follows:

- Selected 9 larger triangular patches {**L3L1L2**, **AAL3L4L3**, **L3CL2**, **L4L2L1**, **L5L3L6**, **L5L2AAL3**, **L5L7AAL6**, **CL2L7**, **CL3L6**}
- Selected 15 smaller triangular patches {**L1CL2**, **L2CM1**, **M1CM2**, **M2CM3**, **L3CM4**, **M4CL4**, **L1CL7**, **M8CM7**, **M6CL6**, **L6CM5**, **M5CL4**, **L4L3AL3**, **L4L5AAL3**, **L4L5AAL6**, **AAL6AAL3L5**}
- Selected 40 nasal curves {**L1L2**, **L1 M1**, **L1 M2**, **L1 M3**, **L1 M4**, **L1 L5**, **L1 L7**, **L1 M5**, **L2 L7**, **L2 C**, **L2 L4**, **L2 AAL3**, **L2 M8**, **L7 M2**, **L7 M1**, **M2 C**, **AAL3 C**, **L4 C**, **L6 C**, **AAL6 C**, **L4 M1**, **L4 M2**, **L4 M3**, **L4 L3**, **L4 AAL3**, **L4 L5**, **L4 AAL6**, **L4 M6**, **L4 M7**, **M4 M2**, **M4 M7**, **M5 M7**, **M5 M2**, **M1 M8**, **M3 M6**, **AL3 AL6**, **AAL3 AAL6**, **AAL3 L5**, **AAL6 L5**, **M7 L3**}

In order to see how the rank-one recognition rates vary for different selected patches and curves by GA, these two factors are plotted against each other in Fig. 8-18. The points are obtained from different iterations of the GA optimiser. These figures provide very interesting information about which combination of regions are less expression sensitive. Also, it is helpful to see which curves and patches are able to simultaneously generate a high recognition rank and low dimensional feature space.



(a)



(b)

Figure 8-16: Feature selection results for different GA's iterations: (a) Curves and spherical patches; (b) Smaller and larger triangular patches.

For these specific points, the corresponding curves and patches are plotted in Fig. 8-19: 1) For 16 selected spherical patches, when $R_1 = 0.9693$; 2) For 10 selected small triangular patches, when $R_1 = 0.9504$; 3) For 5 selected large triangular patches, when $R_1 = 0.9358$; 4) For 29 selected curves, when $R_1 = 0.9581$. The feature selection results show that the landmarks for the spherical patches and curves are mostly concentrated on the eye corners area and nose tip. On the other hand, their distribution is less dense on the nasal cartilaginous sides

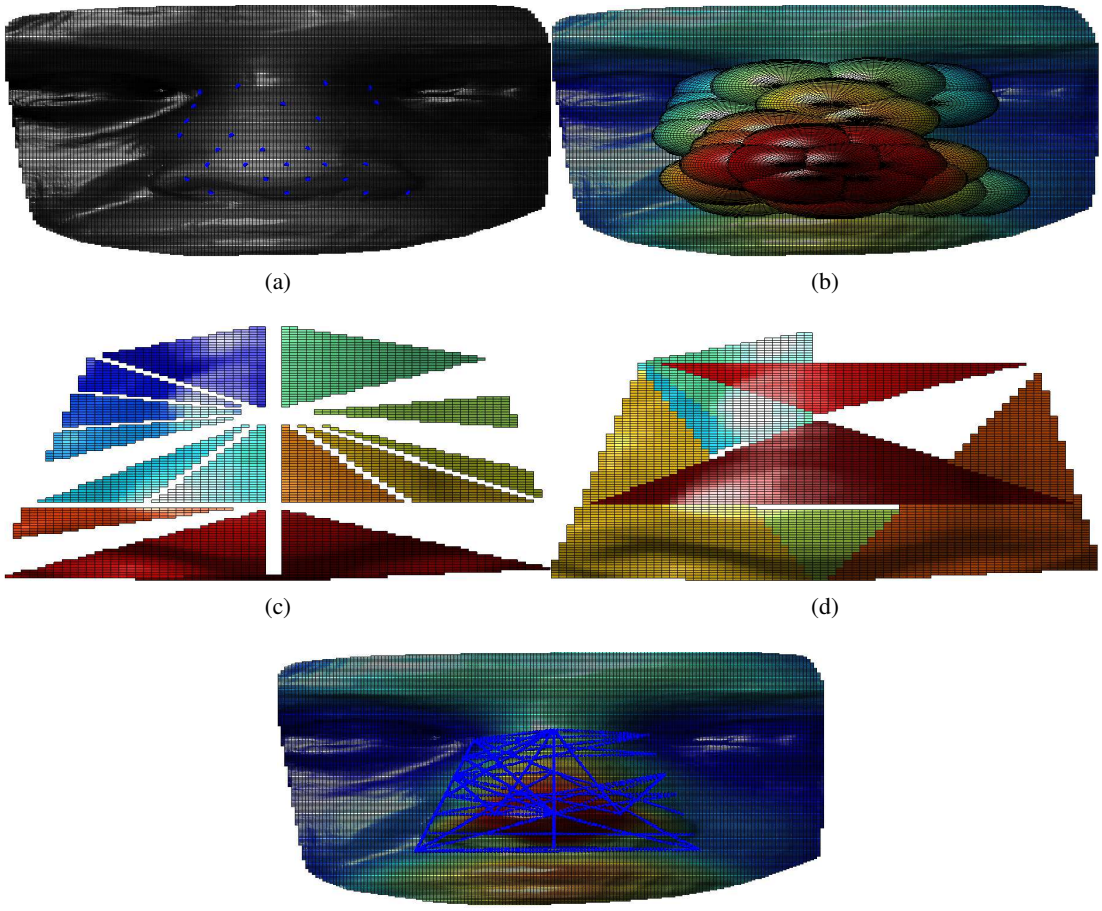


Figure 8-17: Feature descriptors corresponding to the selected features for: (a and b) Spherical patches; (c) Small triangular patches; (d) Large triangular patches; (e) Nasal curves.

parts and the dorsal side parts having expression sensitive muscles, which are less stable in facial expressions. This is also can be seen for the smaller triangular patches, as the nasal alar parts and dorsal side walls are omitted by the feature selector.

On the contrary, the feature selection Fig. 8-19-d shows that for the larger triangular patches, if a combination of regions including the nasal alar, dorsal and tip is used for the feature space, it will be the least sensitive to expressions. However, the previous regions have also been multiply detected (the nose tip and eye corners near the dorsal area), which again shows these regions higher stability in different expressions.

Finally, Fig. 8-20-a illustrates the recognition ranks before and after applying the feature selection on the Bosphorus dataset. There is an average improvement about $\approx$1% for all the modalities. The spherical patches produce te highest recognition performance, while the larger

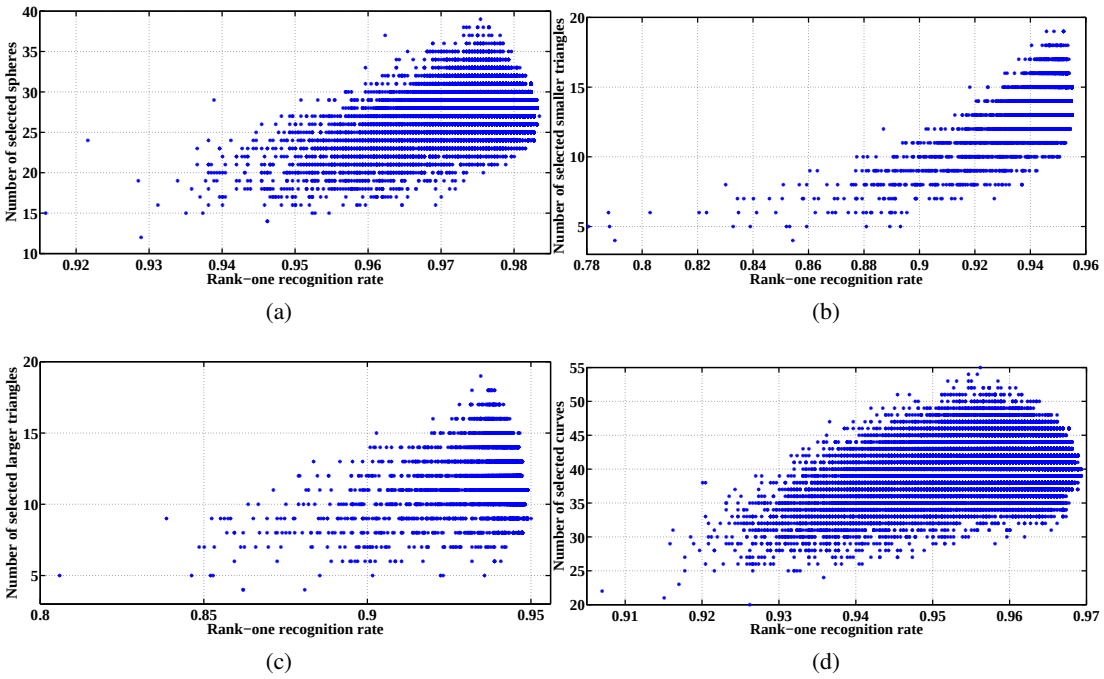


Figure 8-18: Number of the selected feature descriptors vs. the rank-one recognition rate for: (a) Spherical patches; (b) Small triangular patches; (c) Large triangular patches; (d) Curves.

triangular patches have the lowest performance for ranks < 6.

Further experiments on Bosphorus

The Bosphorus dataset contains subjects with various expressions types. Two sets of experiments are defined and performed on the Bosphorus dataset to evaluate the proposed method's robustness for an expression-robust 3D nose recognition. The first experiment is to use a gallery consisting of one neutral training sample only. Therefore, for 105 subjects in the dataset, 105 samples are used as the gallery and the rest of non-neutral and neutral samples as probe. The cumulative matching characteristic (CMC) curve for this experiment is depicted in Fig. 8-20-b. As seen in the previous results, the spherical patches produced the highest recognition ranks, starting at 95.28% and quickly reaching 99% for ranks > 4.

This set up has been utilised in previous researches' experiments in the literature. The comparative results are depicted in Table. 8.3. There has been just one previous approach on the nasal region, which has used nasal region recognition on the Bosphorus dataset, while the approach has a significant better performance even a larger dataset is used. Compared to some other algorithms, the spherical patches results are still comparative although the nasal region is utilised.

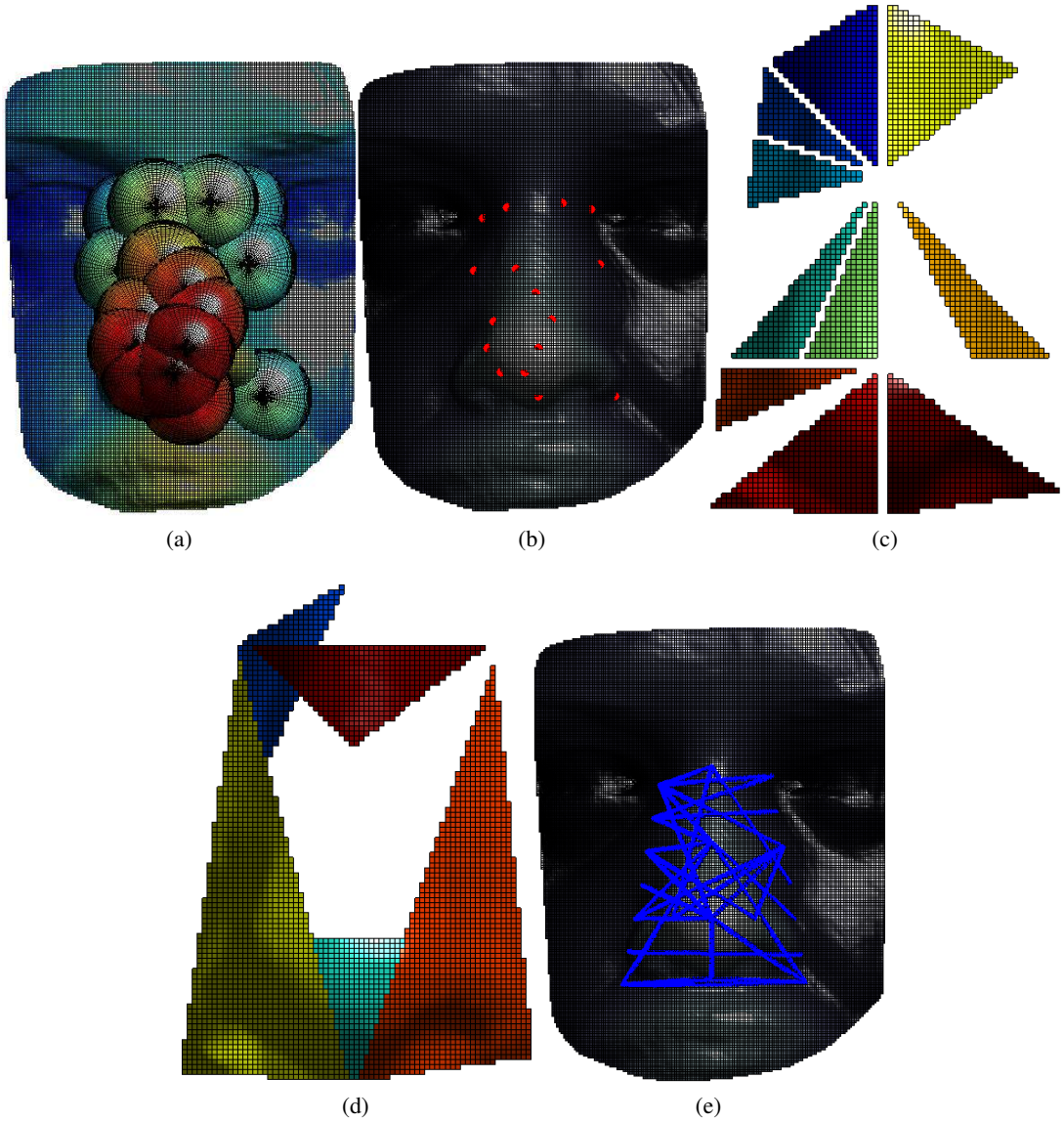
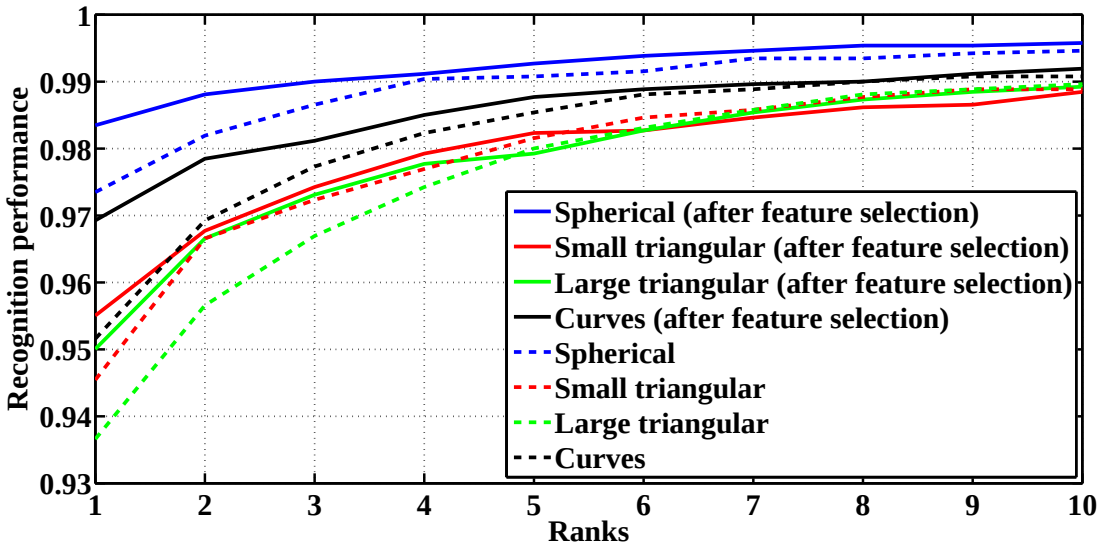
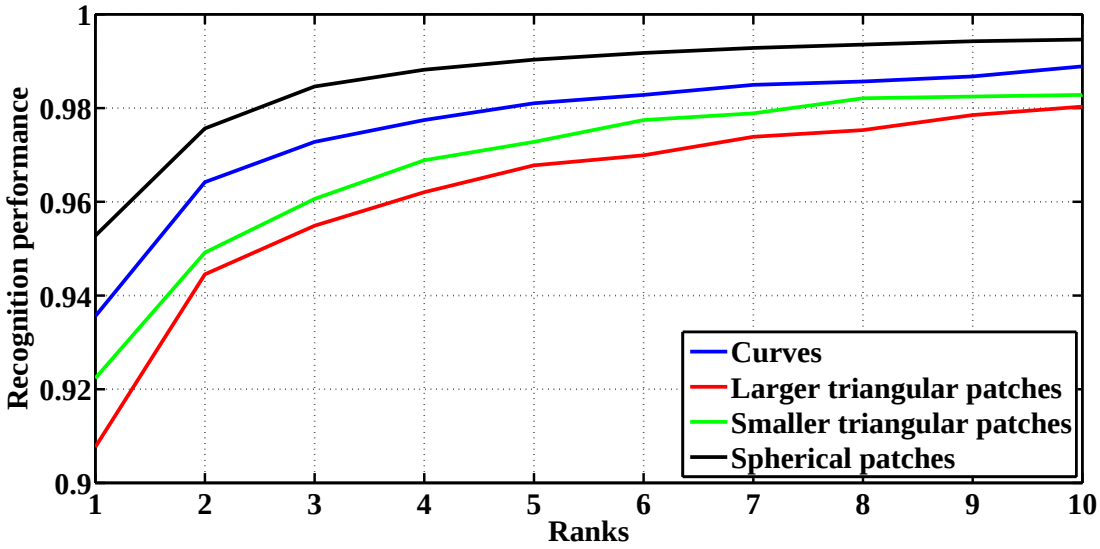


Figure 8-19: Examples of how a few combination of feature descriptors can have high discriminatory strength: (a and b) 16 spherical patches $R_1 = 0.9693$; (c) 10 smaller triangular patches $R_1 = 0.9504$; (d) 5 larger triangular patches $R_1 = 0.9358$; (e) 29 nasal curves $R_1 = 0.9581$.

The second experiment is based on using various expressions for the probe samples. The dataset includes neutral, anger, disgust, fear, happiness, sadness, and surprise expressions. In order to see the effects of utilising incompatible expressions for the gallery as compared with the probe, for each subject, an expression is selected for the gallery samples and another expression for the probe. The results can be demonstrated in a 1×7 block matrix (Table 8.4), where each column relates to an expression. There are 76 subjects with more than one neutral



(a)



(b)

Figure 8-20: (a) CMC curves for after and before feature selection on the Bosphorus dataset for neutral gallery vs. non-neutral probe. (b) CMC curve on the Bosphorus dataset for one neutral sample per subject gallery (105 samples) vs. the 2797 other samples as probe.

samples per subject. For those, the average recognition rank is reported for the neutral vs. neutral experiment (the last column).

For each experiment, the number of samples used in the probe set is also mentioned. The spherical patches outperforms other feature descriptors except for the fear, in which the curves and smaller triangular patches have better rank-one recognition rates. The disgust expression deforms the noses more significantly than other expressions as the lowest recognition ranks are

Algorithm	Modality and size	Rank one
Spherical Small tri Large tri Curves	3D Nose (105/2797)	**95.28%** 92.23% 90.77% 93.56%
Li *et al.* (2014) [85]	3D Face (105/2797)	95.4%
Dibeklioğlu [140]	3D Nose (47/1527) (47/423) → rotation	 89.2% 62.6%
Li *et al.* (2011) [160]	3D Face (105/4561)	 94.1%
Alyüz *et al.* (2010) [58]	3D Face (105/2814)	 **98.2%**
Alyüz *et al.* (2008) [161]	3D Face (34/441) (47/1508)	 95.87% 95.29%

Table 8.3: Comparison of some of the previous works on the Bosphorus dataset. The number of samples used for training and test are shown in brackets.

obtained for this type of expression. On the other hand, the feature space is mostly remains invariant for the happy and surprise expression, as 100% of the samples are recognised correctly for these probe sets. The algorithm is also compared with the one proposed by [160], which used similar evaluations.

Variable training size for FRGC

Increasing the number of training samples per class should reasonably increase the probability of correct matching and as a consequence, the recognition ranks. In order to see the effects of using different training samples per subject, the training size is enlarged for each person in the FRGC dataset and the rank-one recognition rates are computed for the selected patches and curves. The results are shown in Table 8.5.

To do this, all the folders in FRGC from different seasons are merged. Then for each subject, the number of samples in the gallery is changed and the average recognition ranks are reported after the samples are interchanged in the gallery and probe. The results show the high discrimination of the feature space for the spherical patches for this well-known dataset. When only one sample per subject is used in the gallery (482 gallery samples vs. 4330 probe samples to recognise 482 subjects), 96.19% rank-one recognition rate is achieved. Compared to the

Algorithm	Probe						
	Happy (106)	**Surprise (71)**	**Fear (70)**	**Sadness (66)**	**Anger (71)**	**Disgust (69)**	**Neutral (194)**
Spherical	100%	100%	98.55%	98.46%	98.53%	94.12%	99.48%
Small tri	96.15%	100%	100%	96.92%	95.59%	86.76%	98.44%
Large tri	93.27%	98.59%	97.10%	96.92%	94.12%	80.88%	98.96%
Curves	97.12%	98.59%	100%	96.92%	92.65%	94.12%	98.96%
Li *et al.* [160]	95.28%	98.59%	92.86%	95.45%	88.73%	76.81%	100%

Table 8.4: Rank-one recognition rate for different expression types from the Bosphorus dataset used as probe, while the neutral samples are used as the gallery.

Feature descriptors	Gallery size per subject						
	1 (482/ 4330)	**2 (880/ 3848)**	**3 (1206/ 3408)**	**4 (1432/ 3006)**	**5 (1610/ 2648)**	**6 (1752/ 2326)**	**7 (1757/ 2034)**
Spherical	96.19%	98.91%	99.38%	99.60%	99.62%	99.70%	99.75%
Small tri	89.66%	95.75%	97.62%	98.19%	98.65%	98.89%	99.08%
Large tri	88.98%	96.28%	97.73%	98.38%	98.80%	98.98%	99.22%
Curves	91.64%	96.78%	98.09%	98.77%	98.94%	99.27%	99.36%

Table 8.5: Increasing the training size per subject, when all samples of the FRGC dataset are merged from the three season folders (at the top of the columns: # of gallery samples/# of probe samples).

results reported by recent research, this is the highest 3D nasal region recognition rank ever obtained from this dataset and comparable with many other 3D face recognition algorithms, which used the whole facial domain. Although other three feature descriptor modalities have lower rank-one recognition rates for one training sample per subject, there is a big increase in their recognition performance when the samples per subjects are increased. For instance, when 2 samples per subject is used in the gallery, the nasal curves rank-one recognition performance increase for more than 5%.

FRGC v2.0 and ROC III

The second experiment is performed on the FRGC v2.0 dataset, which is a widely used face recognition benchmark. The dataset contains various samples with varying expression for both the gallery and probe (although not as diverse as the Bosphorus). The recognition performance for other ranks when FRGC v2.0 is used, is also illustrated in Fig. 8-21-a. For rank > 2, the

recognition performance increases to $> 99\%$ for the spherical patches. Similar trend exists for the other modalities' CMC curves, as they all increase with a high gradient when ranks > 2.

Algorithm	**Modality**	**Rank-one FRGC v2.0**	**EER ROC III**	**0.1% FAR ROC III**	**Neutral vs. Neutral**	**Neutral vs. Non-neutral**
Spherical Small tri Large tri Curves	3D Nose	**97.87%** 92.87% 92.77% 94.06%	2.4% 4.9% 6.4% 4.9%	93.5% 83.3% 85.3% 80.0%	**98.45%** (Rank1) 95.09% (Rank1) 95.68% (Rank1) 95.77% (Rank1)	**98.51%** (Rank1) 96.31% (Rank1) 96.57% (Rank1) 97.49% (Rank1)
Smeets *et al.* [159]	3D Face	89.6%	3.8%	77.2%	-	-
Osaimi *et al.* [108]	3D Face	96.5%	-	94.05%	98.35% (0.1% FAR)	97.8% (0.1% FAR)
Spreeuwers *et al.* [88]	3D Face 3D Nose	**99.0%** 94.5%	-	94.6% 83.7%	-	-
Drira *et al.* (2013) [42]	3D Face	97.0%	-	97.1%	99.2% (Rank1)	96.8% (Rank1)
Alyüz *et al.* (2010) [58]	3D Face 3D Nose	97.5% 91.81	1.91% -	85.6% -	98.39% (Rank1) -	96.40% (Rank1) -
Wang *et al.* (2010) [36]	3D Face	98.39%	-	98.04%	99.2% (0.1% FAR)	97.7% (0.1% FAR)
Wang *et al.* (2008) [89]	3D Nose	95% (44mm) 78% (24mm)	-	-	-	-
Drira *et al.* (2009) [95]	3D Face/Nose (125 gallery) (125 probe)	88% (Face) 77.5% (Nose)	-	-	-	-
Chang *et al.* [67]	3D Nose	-	Neutral 12% Non-neutral 23%	-	97.1% (RANK1)	86.1% (RANK1)
Emam-bakhsh *et al.* [2]	3D Nose	89.61%	Neutral 8% Non-neutral 18%	-	90.87% (RANK1)	81.61% (RANK1)
Li *et al.* (2014) [85]	3D Face	96.3%	-	-	98.0% (RANK1)	94.2% (RANK1)
Queirolo *et al.* [57]	3D Face	**99.6%**	-	96.6%	99.5% (RANK1)	94.8% (RANK1)
Berretti *et al.* (2013) [97]	3D Face	95.6%	-	-	97.3% (RANK1)	92.8% (RANK1)
Berretti *et al.* (2010) [99]	3D Face	94.15%	-	-	≈97.3% (RANK1)	≈91.0% (RANK1)
Moham-madzade *et al.* (2013) [38]	3D Face	-	-	**99.2%**	-	-
Mian *et al.* (2008) [70]	3D Face	93.5%	-	Neutral 99.9% Non-neutral 92.7%	99%	86.7%
Mian *et al.* (2007) [32]	2D+3D Face 2D+3D Nose	95.91% ≈92.2%	-	99.3% 92.5%	99.2% ≈94.9%	95.37% ≈80.0%

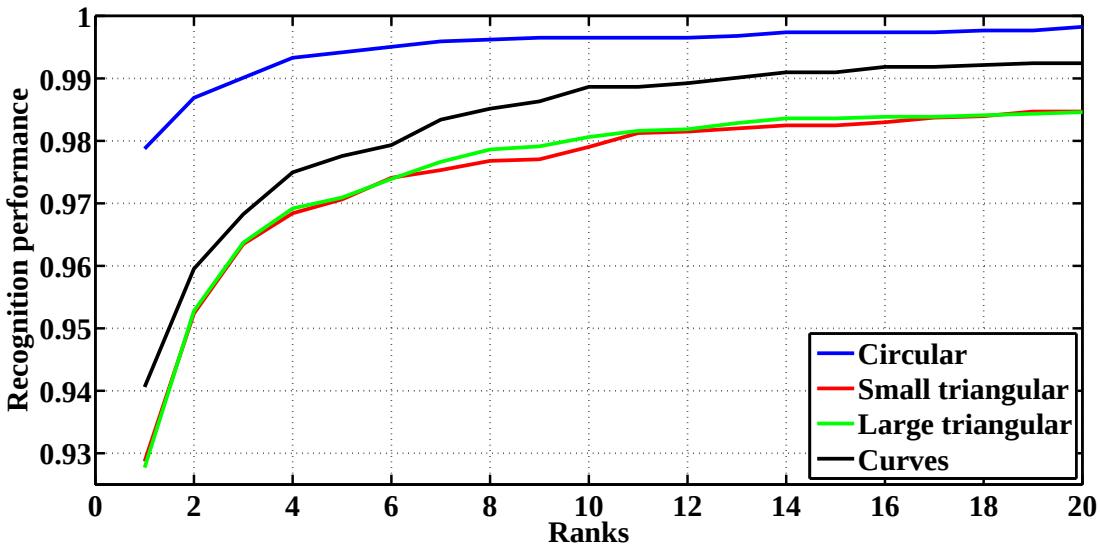
Table 8.6: Comparison of the results on the FRGC dataset.

The result of applying different patches and curves on this dataset is illustrated in Table 8.6. The table compares the performance of the proposed approach with recently proposed 3D face recognition techniques that use the nasal region and also the whole face. For the FRGC v2.0 experiment, the spherical patches outcome outperforms other feature descriptors modalities with a 97.87% rank-one recognition rate. For this experiment, although only the nasal region is used, the proposed algorithm's recognition rates are higher than or very close to the results of [159, 108, 42, 58, 89, 85, 97, 32], in which the whole facial surface is utilised. Moreover, the nasal region recognition performance of [88, 58, 89, 32] are also $\approx$ 4%, 6%, 3% and 5% lower than the spherical patches'. Also, when the samples used in the probe set are changed from neutral to non-neutral, the proposed algorithm has the lowest variation of the recognition rate for all the feature descriptors ($\approx 0.06\%$ for the spherical descriptors). An intersting fact is that there is even a slight increase in the recognition rates, when non-neutral samples are used as probe (the last column). Furthermore, in terms of the similarity of the feature extraction methodology to the proposed method, Li *et al.* [85] also used normal vectors of the face with a weighted sparse classifier. As illustrated in Table. 8.6, there is an approximate 4% decline when non-neutral samples of FRGC are used as probe. This difference in the recognition rate is significantly higher than all of the proposed feature descriptors.

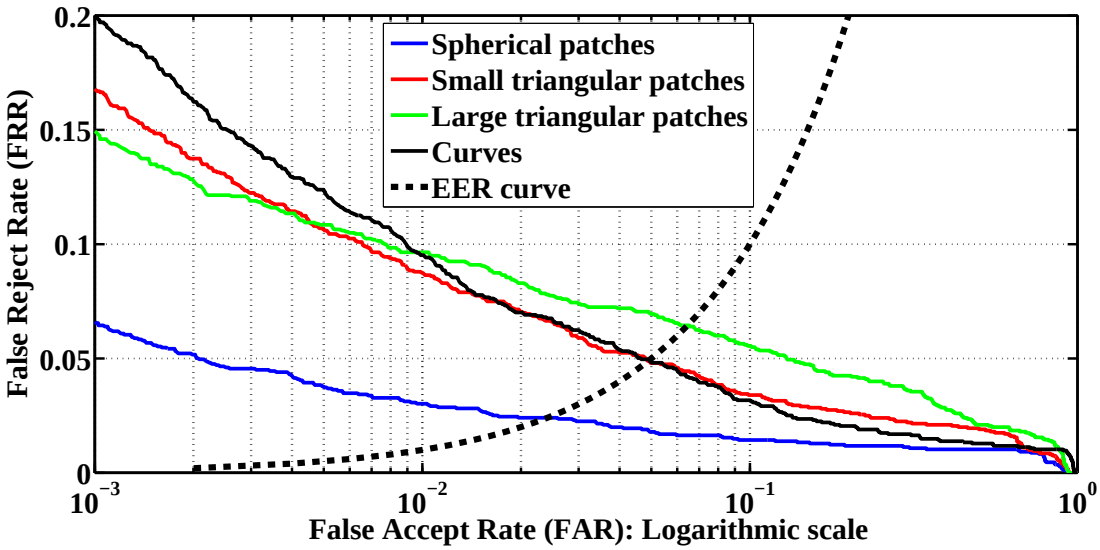
ROC III, which is the FRGC's cross-seasonal verification scenario is also implemented on the 3D nose recognition system. The ROC III curve is plotted in Fig. 8-21-b in a logarithmic scale. The EER and 0.1% FAR for the proposed approach are 2.4% and 93.5%, respectively. Although they are still higher than previous 3D nose recognition algorithms and some 3D face recognition methods, as can be seen from the fifth column in Table 8.6, some of the algorithms that used the whole facial domain shows more robustness for the verification scenario. To be more specific, for some of the algorithms such as [108, 70, 42, 88], which have lower performance for the identification scenario, the verification performance outperforms the proposed nasal curves and patches. It might be showing that for the verification scenarios, the whole facial region might provide a higher confidence level to be matched with a claimed identity.

8.7 Disscussion and future work

The objective function could be improved to consider other expression types for the gallery, as well as the neutral samples. The current objective function only assumes that the gallery only contains the neutral expression samples. This configuration is assumed because it has been used in many previous published research and to be able to make comparison with the results in the literature. However, in a real-world application, the gallery samples can include any arbitrary expression types. The performance of the current algorithm for different expressions



(a)



(b)

Figure 8-21: The result of verification and identification scenarios, when nasal curves, spherical and triangular patches are used over the FRGC v2.0 dataset: (a) CMC curves for FRGC v2.0; (b) Between seasons verification results for FRGC: ROC III.

are depicted in Tables 8.7 and 8.8, for the Bosphorus and FRGC v2.0 datasets, respectively.

The Bosphorus dataset includes neutral, anger, disgust, fear, happiness, sadness, and surprise expressions. In order to see the effects of utilising incompatible expressions for the gallery and probe, for each subject, a expression is selected for the gallery samples and another expression for the probe. The results can be demonstrated in a 7×7 block matrix, where each row and column relates to an expression. There are some subjects in the Bosphorus dataset,

Feature descriptors	Gallery	Probe						
		Happy	**Surprise**	**Fear**	**Sadness**	**Anger**	**Disgust**	**Neutral**
Spherical	**Happy**	N/A	86.96%	76.12%	74.60%	74.24%	83.33%	91.50%
Small tri			78.26%	70.15%	82.54%	74.24%	72.73%	88.44%
Large tri			75.36%	71.64%	76.19%	66.67%	75,76%	84.01%
Curves			81.16%	74.63%	74.60%	66.67%	78.79%	86.39%
Spherical	**Surprise**	84.29%	N/A	97.10%	90.77%	89.71%	77.94%	97.53%
Small tri		81.43%		92.75%	87.69%	83.82%	70.59%	95.68%
Large tri		71.43%		95.65%	90.77%	83.82%	70.59%	93.83%
Curves		80.00%		95.65%	89.23%	83.82%	77.94%	93.83%
Spherical	**Fear**	80.88%	100%	N/A	100%	93.94%	84.85%	96.88%
Small tri		72.06%	94.20%		96.83%	92.42%	72.73%	93.13%
Large tri		72.06%	94.20%		96.83%	83.33%	72.73%	90.00%
Curves		73.53%	97.10%		96.83%	90.91%	80.30%	93.13%
Spherical	**Sadness**	84.13%	93.85%	98.41%	N/A	96.83%	89.06%	96.64%
Small tri		74.60%	95.38%	98.41%		92.06%	81.25%	95.30%
Large tri		73.02%	87.69%	96.83%		92.06%	73.44%	94.63%
Curves		79.37%	89.23%	98.41%		90.48%	85.94%	95.30%
Spherical	**Anger**	80.60%	86.76%	92.42%	95.24%	N/A	86.15%	90.38%
Small tri		67.16%	85.29%	89.39%	92.06%		78.46%	88.46%
Large tri		68.66%	85.29%	84.85%	90.48%		84.62%	88.46%
Curves		71.64%	82.35%	90.91%	90.48%		86.15%	87.82%
Spherical	**Disgust**	81.82%	72.06%	78.79%	79.69%	87.69%	N/A	85.99%
Small tri		65.15%	64.71%	71.21%	73.44%	73.85%		78.34%
Large tri		77.27%	63.24%	69.70%	76.56%	75.38%		77.71%
Curves		74.24%	70.59%	74.24%	78.13%	78.46%		78.98%
Spherical	**Neutral**	100%	100%	98.55%	98.46%	98.53%	94.12%	99.48%
Small tri		96.15%	100%	100%	96.92%	95.59%	86.76%	98.44%
Large tri		93.27%	98.59%	97.10%	96.92%	94.12%	80.88%	98.96%
Curves		97.12%	98.59%	100%	96.92%	92.65%	94.12%	98.96%

Table 8.7: The expression vs. expression experiment for the Bosphorus dataset. The neutral and non-neutral samples in the Bosphorus dataset are used as both the gallery and probe, and the rank-one recognition rate is computed.

which does not have some specific expression types. Although this is not a common issue for most of the subjects, these samples are removed are removed from the test samples, because they do not possess any corresponding sample in the gallery. Also, for the case in which similar expression types are compared (the diagonal elements of Table. 8.7), one sample is used in the gallery and the rest of the subject's samples as probe. Nearly all of the subjects in the dataset have only one sample with non-neutral expression. This is the reason why the diagonal elements are all N/A in the table. However, there are 76 subjects with more than one neutral samples. For those subjects the average recognition rank is reported.

The lowest recognition rank is for the case, in which the disgust expression type is used as probe. The similar situation exists, when the neutral set is used as the probe (last column). The gallery including the fear expression samples also causes the least deformations on the

nasal region, as can be seen from the fourth row. Moreover, using the table, one can see which expression pairs result in a more alike deformation on the nasal region. For instance, a gallery consisting of sadness expression, produces a projected feature space highly more similar to the fear expression (98.41% rank-one recognition). However as it might be expected, the sadness gallery is the least alike to the happy probe set. As another example, the noses in a probe set with anger expression, have the lowest similarity with the happy gallery (the second row and seventh column). On the other hand, the same probe set is significantly more correlated with the sadness gallery (same column, the fifth row).

The same issue exists for the FRGC dataset as demonstrated in table 8.8, i.e when non-neutral samples are used for the gallery the output recognition performance is degraded. One way to solve this issue, would be to modify the objective function in a way to handle all the possible combinations of expression types as the gallery and probe. finding the optimal point of such function would not only make the algorithm robust against non-neutral probe, but also, against non-neutral gallery as well.

Feature descriptors	**Gallery**	**Probe**	
		Neutral	**Non-neutral**
Spherical	**Neutral**	98.45%	98.51%
Small tri		95.09%	96.31%
Large tri		95.68%	96.57%
Curves		95.77%	97.49%
Spherical	**Non-neutral**	87.10%	94.51%
Small tri		85.68%	89.78%
Large tri		85.58%	87.84%
Curves		85.64%	89.75%

Table 8.8: The expression vs. expression experiment for the FRGC dataset. The neutral and non-neutral samples in the FRGC dataset are used as both the gallery and probe, and the rank-one recognition rate is computed.

8.8 Conclusion

To tackle the problem of expression invariant face recognition, a novel algorithm is introduced to utilise the 3D shape of nose. The algorithm is based on a highly consistent and accurate landmarking algorithm, a robust feature space, discriminative feature descriptors and feature selectors. The proposed method is applied on the popular face datasets, FRGC and Bosphorus. The matching results show very successful performance of the nose region for both identification and verification scenarios. The outcome is 97.87% rank-one rate on FRGC v2.0, 2.4%

EER on ROC III, and 98.45% and 98.51% for neutral and non-neutral probes, respectively. The proposed method does not rely on sophisticated preprocessing algorithms, such as denoising and alignment. Also, when there is only one sample per subject in the gallery, for all the merged folders of the FRGC dataset, the rank-one recognition rate is 96.19%. The results obtained from the proposed method reveal the high potential of the nasal region for face recognition. The recognition ranks is not only significantly higher than previous nasal region-based algorithms, but also has better performance than 3D holistic and multi-modal approaches.

There are several aspects of the algorithm, which can be utilised in other applications. To be more specific, the feature extraction step, which is based on the histograms of Gabor wavelet normals can be applied to other 3D object recognition method. Also, the feature selection paradigm explained in this chapter can be easily applied to other pattern recognition algorithm, to maximise the within-class and between-class similarity and dissimilarity, respectively, to be able to extract a lower dimensional and less redundant feature space. Moreover, the application of the feature extraction step on the whole facial region, to make it robust against occlusion can also be an interesting field of research. Furthermore, ulitising the deep learning-based approaches on the feature space can be an alternative for the feature selector, as deep learning has shown high robustness in recent researches.

Chapter 9

Conclusions

9.1 Summary

This thesis has provided a detailed investigation of the preprocessing algorithms and the potential of the 3D nasal region for recognition purposes. The optimal parameters for the widely used denoising algorithms, which are usually applied as a preprocessing step, are found. Then, a novel method is used to learn the noise distribution over the 3D faces and to simulate the noise over any arbitrary face. A novel nose tip detection approach is also proposed, which is robust against the curvature maps thresholding and can detect the nose tip for various poses and expressions. In addition, the nasal region's application to align faces having different expressions and partial/self-occlusions is introduced. The two final experimental chapters of this thesis are focused on the nasal region's potential for 3D face recognition. Two algorithms are explained, which are based on the nasal curves, computed from the depth maps, and patches and curves calculated by the normal maps of the Gabor wavelet filtered depth maps. The algorithms, which are both based on robust landmarking approaches, confirm the very high discriminant power of the nose region for expression-robust 3D face recognition. In the following subsections, a summary of each chapter is provided.

9.1.1 Denoising evaluation on 3D face recognition methods

Some widely reconised denoising methods used for preprocessing in many popular 3D face recognition algorithms were analysed. These included, mean, median, Gaussian filtering, the statistical Weiner filtering, non-linear diffusion and various wavelets. A widely used 3D

face recognition pipeline, which consists of denoising, face cropping, alignment, normalisation and resampling, is then combined with the recognition algorithms. The holistic matching/classification methods used in this work were the multi-class SVM, Eigenfaces, Fisherfaces, KFA, PNN, KNN and TreeBagger algorithms. The main purpose of the experiments were to: 1) detect the most optimal parameters for the denoising algorithms to be used for 3D face recognition methods; 2) detect the most vulnerable holistic 3D face recognition methods (among the ones used) to noise; and finally, 3) which denoising algorithm has the best performance when applied as a preprocessing step for 3D face recognition algorithm. The results were were obtained using FRGC's noisiest folder (Spring 2003). The results show that while KFA had the highest rank-one recognition rate, the median filtering had a significantly higher performance than other denoising algorithms. A very interesting outcome of the results was that intensive denoising, which refers to utilising large masks and producing blurrier results, does not necessarily reduce the recognition rates. This conclusion contradicts the initial hypothesis that too blurry facial depth image might lose its between-class dissimilarity, since the edges would be lost. The results, however, showed that larger masks can not only maintain but also improve the descrimination of 3D faces.

A novel noise modelling and simulation algorithm was also proposed, to explore the robustness of denoising and 3D holistic face recognition algorithms for different noise powers. This enables a quantitative evaluation of the robustness to noise of different classification and denoising methods to be performed. To do this the novel approach of learning the noise distribution over the facial surface and then simulating it over other samples is proposed. Median, Gaussian and Weiner filtering generate the best results, with the median filter producing the best overall performance for median to high noise intensity. For low intensity noise, subspace projection classifiers (KFA and LDA) outperform the other methods. However, when the noise intensity is increased, the performance of subspace projection methods significantly deteriorates and, in the experiments, the TreeBagger and KNN with the city-block distance show the best robustness. The use of denoised/noisy samples for the gallery/probe is also evaluated. These results show that matching algorithms (KNN-Ctb) significantly outperform the training-based methods, as they do not rely on classification boundary allocation or subspace projection.

9.1.2 Nose tip detection algorithm

A significant number of 3D face recognition algorithms rely on the nose tip detection for facial region segmentation. This motivated the development of a new 3D nose tip detection method, which does not have the vulnerabilities of traditional curvature-based thresholding. The proposed method was based on defining a range of thresholding bands over the SI map, instead of

fixed thresholds. Then, using the fact that the nose has a significant consistency in its convexity over different yaw and pitch directional rotations, a new transform was defined, to map the 2D real domain input depth data to a binary space, known as the **IFill** operator. The histogram of an energy function defined using the **IFill**'s output was used to detect a set of candidate points for the nose tip. A mixture of Gaussian function are then used to create a heat map. Because of the nasal tip regions' superior convexity compared to other facial parts, a salient peak is produced on the nose tip location over the heat map.

The nose tip detector algorithm was tested over the Bosphorus dataset, which includes significant number of samples with different expressions and pose variations. The algorithm was evaluated by comparison with the ground truth. Precision curves showed around 90% correct detection in the range of $7mm$ from the ground truth. Also, the algorithm is robust against yaw self-occlusion, up until at least 45°rotations. In summary, the proposed approach provides a capability to overcome the issues involved with the previous direct curvature thresholding and is able to detect the nose tip from different expressions and self-occluded faces.

9.1.3 Alignment algorithms using the nasal region

Face image alignment is an essential step for most of the 3D face recognition algorithms and is analysed in chapters 5 and 6. PCA, which has been widely utilised for 3D face pose correction, is sensitive to partial and self-occlusions. To address this issue, novel algorithms were proposed for 3D face alignment using the nasal region, based on the **IFill** operator. The algorithm is explained in chapter 5 but, is sensitive to occlusion and performs the alignment in two steps by optimising two different objective functions. In order to make it robust against occlusion and unify the alignment procedure in a single step, further 3D pose correction approach is introduced in chapter 6 and applied to three face datasets, FRGC, Bosphorus and UMB-DB, which include different expressions, small and large rotations (with provided ground truth) and various partial occlusion. The algorithm's consistency was quantified using the ICP error and the outcome showed significant improvement for the partially occluded and self-occluded samples alignment compared to PCA.

9.1.4 Nasal curves matching

The 3D nasal region's potential for face recognition was analysed in chapters 7 and 8. The algorithms were based on consistent and accurate nasal landmarking approaches. In chapter 7, four basic landmarks were localised on the nose: the nose tip, root, and left and right alar groove. Then these basic landmarks were used to localise some other set of keypoints on the nasal region. All of the resulting landmarks were utilised to create a set of nasal curves over

the depth map. The feature space was then generated by concatenating the curves. To make the algorithm more robust against the deformations caused by expression, an objective function was used to find the optimal set of robust nasal curves. For this purpose, FSFS and a genetic algorithm were used. The results, which were obtained over the FRGC and Bosphorus datasets, showed significant potential of the proposed method and nasal region to be used for 3D face recognition. Various classification algorithms were also analysed over the feature space. KFA with the polynomial kernel generated the highest rank-one recognition rate, while its resulting EER were approximately 4% lower than the best previously 3D nose recognition algorithm.

9.1.5 Introducing patches for expression-robust 3D nose recognition

Chapter 8 improved the recognition performance of the nasal region by extending the nasal curves to various patches and more curves. First, an even more accurate landmarking algorithm was introduced, which detects the nose tip, root, alar groove, eye corners and subnasale. These landmarks provide the opportunity to incorporate a larger nasal region, than the previous curve-based approach, which lost some parts (such as the lower nasal region and sides of the nose). Instead of utilising the depth map for the feature space, a new feature space based on surface normal vectors was introduced. Gabor wavelets were applied to generate a multi-resolution set of feature descriptors for a set of curves, triangular and spherical patches. A genetic algorithm-based feature selector was used to select the most robust set of features against the variations of expressions.

The algorithm was evaluated on the FRGC and Bosphorus datasets. The quantitative evaluation of the landmarking method showed its remarkable consistency, providing a very low average Eculidean distance displacement error. The 3D face recognition performance evaluated over the FRGC and Bosphorus datasets, showed the proposed algorithm's highest nasal region recognition ranks compared to the previous research. The results were not only higher than previous 3D nose region-based methods, but also higher than or very close to previous algorithms that used the whole 3D face.

9.2 Discussion and future work

The algorithms proposed in this thesis could be improved in a number of different aspects. Some examples are given in the following subsections.

9.2.1 Denoising algorithm's parameters and type

The denoising algorithm explained in chapter 3 was applied on the holistic methods. The regional, landmark- and curve-based methods, however, might perform differently from holistic methods in the presence of noise. Some parts of the facial surface are intrinsically more difficult to reconstruct in 3D. These parts include the high frequency regions such as the nostrils, nose sides, eyes and mouth corners. Therefore, the effects of changing the denoising methods parameters for these regions could be influential on the overall 3D face recognition performance.

On the other hand, the landmarking methods can be affected, in a different way, by the noise. The accuracy and consistency of landmark detection can be degraded by the noise. To solve this issue, the best denoising method might differ from those for the holistic algorithms. For the curve- or contour-based approaches, the denoising might affect the normalisation step and therefore, the denoising might be applied on each curve instead of the whole facial domain. Considering these challenges, the type of denoising algorithm and/or its parameters is highly dependent on the type of 3D face recognition approach.

9.2.2 Occlusion for the 3D nose recognition

The nose recognition algorithms explained in chapters 7 and 8 used the nasal region robustness to be less sensitive to occlusions caused by hair, moustache and scarf. However, if a subject intentionally occludes some part of his nose, the algorithm could be modified to detect the manipulation on the facial surface and exclude these regions from the feature space. Also, the robustness of the proposed method against occlusion should be quantitatively evaluated. The problem might not be significant for self-occlusion, since a successful alignment algorithm can identify self-occluded regions and they can be labelled and removed from the feature space. However, the amount of self-occlusion that still gives an acceptable recognition rate for the patches and curves can be quantified.

9.2.3 Verification scenarios and non-neutral samples for the gallery

The objective function used for feature selection in chapter 8 only utlises the rank-one recognition rate. This might be reason why the identification rates were significantly higher than the verification performance, such as the EER or 0.1% FAR. If the function is modified in way that both identification and verification parameters are included, the performance of the parameters computed for the verification scenario, computed over the ROC curves, can be increased as well. On the other hand, the objective function should maintain the performance of the current identification rates.

The current objective function is also minimised over the neutral samples as the gallery. Although this has been due to the widely used neutral gallery vs. non-neutral probe experiment in recent published research, the objective function still can be improved to detect the nasal regions, which are most robust, even if non-neutral samples are used in the gallery. The objective function can be modified to quantify the recognition performance, when different expression types are used as the gallery and probe.

9.2.4 Applications of the 3D nasal region for ”soft biometrics”

The feature spaces introduced in chapters 7 and 8, could be used to perform soft biometrics algorithms on the face. For instance, the feature space can be used to evaluate the nasal region's potential for expression type classification. Another example would be to asses the 3D shape of nose for gender classification. It has not been proven that all noses can be specifically categorised into a predefined class. A very interesting application of the proposed feature spaces would be to check if the feature space could be used to classify the shape of noses (for example, Greek, Nubian, Roman, Turn-up or Hawk [40]). These can all be applied for the 3D face recognition algorithms to reject invalid samples very quickly and increase the performance. The existing feature selection method could be modified by simply changing the objective function, based on the classification criterion.

The proposed 3D nose recognition algorithms all have an obvious vulnerability to surgery, similar to any type of biometrics algorithm. In order to determine whether to used or exclude the nasal region for face recognition, automatic algorithms to detect nasal surgery could be used [162].

9.2.5 Low resolution datasets

It could be interesting to analyse the algorithm's sensitivity to low resolution 3D samples or the approximate 3D samples obtained from 2D images. This is mainly because, although the 3D laser scanners have become cheaper during the last decade, many of the available datasets are still in 2D. Therefore, one widely applicable algorithm would be to make the algorithms dataset independent and extend the current 3D face recognition methods to other types of datasets.

List of publications arising from this thesis

Published papers

M. Emambakhsh, J. Gao, and A. Evans, "An evaluation of denoising algorithms for 3D face recognition," in Proceedings of IET 5^{th} International Conference on Imaging for Crime Detection and Prevention (ICDP 2013), 2013.
M. Emambakhsh, A. Evans, and M. Smith, "Using nasal curves matching for expression robust 3D nose recognition," in Proceedings of IEEE 6^{th} International Conference on Biometrics: Theory, Applications and Systems (BTAS 2013), Washington DC, USA, 2013.
M. Emambakhsh and A. Evans, "Self-dependent 3D face rotational alignment using the nose region," in Proceedings of IET 4^{th} International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011), 2011.
J. Gao, M. Emambakhsh, and A. Evans, "A low dimensionality expression robust rejector for 3D face recognition," in Proceedings of IEEE 22^{nd} International Conference on Pattern Recognition (ICPR 2014), Stockholm, Sweden, 2014.

Under review

An invited extended version of my IETs International Crime Detection and Prevention (ICDP) 2013 paper to the IET's Computer Vision journal:
M. Emambakhsh, J. Gao, and A. Evans, "Noise Modelling for Denoising and 3D Face Recognition Algorithms Performance Evaluation," submitted to IET Computer Vision.

In preparation for submission

The algorithm of nasal landmarking and use of patches and curves for expression-robust 3D face recognition:
M. Emambakhsh and A. Evans, "Histogram of Localised Gabor Wavelet Normals on the Nose for One Training Sample per Subject Expression Robust 3D Face Recognition," final preparations to be submitted to IEEE Transaction on Pattern Analysis and Machine Intelligence.

References

[1] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[2] M. Emambakhsh, A. Evans, and M. Smith, "Using nasal curves matching for expression robust 3D nose recognition," in *6th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6, 2013.

[3] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[4] A. Jain, A. Ross, and S. Pankanti, "Biometrics: a tool for information security," *IEEE Transactions on Information Forensics and Security*, vol. 1, pp. 125–143, 2006.

[5] R. O. Duda, P. Hart, and D. Stork, *Pattern classification*. John Wiley and Sons, 2001. Page 15.

[6] D. Zhang and W. Zuo, "Computational intelligence-based biometric technologies," *Computational Intelligence Magazine, IEEE*, vol. 2, pp. 26–36, 2007.

[7] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351–359, 2010.

[8] K. Moustakas, D. Tzovaras, and G. Stavropoulos, "Gait recognition using geometric features and soft biometrics," *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 367–370, 2010.

[9] M. Goffredo, I. Bouchrika, J. Carter, and M. Nixon, "Self-calibrating view-invariant gait biometrics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 997 –1008, 2010.

[10] R. Seely, S. Samangooei, M. Lee, J. Carter, and M. Nixon, "The university of Southampton multi-biometric tunnel and introducing a novel 3D gait dataset," in *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6, 2008.

[11] W. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.

[12] K. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition," *Computer Vision and Image Understanding*, vol. 101, pp. 1–15, 2006.

[13] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," in *Biometrics and Identity Management*, vol. 5372, pp. 47–56, Springer Berlin / Heidelberg, 2008.

[14] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 947– 954, 2005.

[15] A. Colombo, C. Cusano, and R. Schettini, "UMB-DB: A database of partially occluded 3D faces," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2113–2119, 2011.

[16] S. Zafeiriou, M. Hansen, G. Atkinson, V. Argyriou, M. Petrou, M. Smith, and L. Smith, "The Photoface database," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 132–139, 2011.

[17] D. Smeets, P. Claes, D. Vandermeulen, and J. Clement, "Objective 3D face recognition: Evolution, approaches and challenges," *Forensic Science International*, vol. 201, pp. 125–132, 1999.

[18] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, pp. 399–458, 2003.

[19] A. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey," *Pattern Recognition Letters*, vol. 28, pp. 1885–1906, 2007.

[20] K. Bowyer, K. Chang, and P. Flynn, "A survey of approaches to three-dimensional face recognition," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, pp. 358–361, 2004.

[21] K. Chang, K. Bowyer, and P. Flynn, "Face recognition using 2D and 3D facial data," in *Workshop on Multimodal User Authentication (MMUA)*, pp. 25–32, 2003.

[22] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *Proceedings of the IEEE*, vol. 83, pp. 705 –741, 1995.

[23] J. Kittler, A. Hilton, M. Hamouz, and J. Illingworth, "3D assisted face recognition: A survey of 3D imaging, modelling and recognition approaches," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 114, 2005.

[24] M. Hamouz, J. Tena, J. Kittler, A. Hilton, and J. Illingworth, "Algorithms for 3D-assisted face recognition," in *14th IEEE Signal Processing and Communications Applications*, pp. 1–4, 2006.

[25] A. Scheenstra, A. Ruifrok, and R. Veltkamp, "A survey of 3D face recognition methods," in *Audio- and Video-Based Biometric Person Authentication*, vol. 3546 of *Lecture Notes in Computer Science*, pp. 325–345, Springer Berlin / Heidelberg, 2005.

[26] M. Emambakhsh, J. Gao, and A. Evans, "An evaluation of denoising algorithms for 3D face recognition," in *5th IET International Conference on Imaging for Crime Detection and Prevention (ICDP 2013)*, pp. 1–6, 2013.

[27] B. Efraty, E. Bilgazyev, S. Shah, and I. A. Kakadiaris, "Profile-based 3D-aided face recognition," *Pattern Recognition*, vol. 45, no. 1, pp. 43–53, 2012.

[28] I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, L. Yunliang, N. Karampatziakis, and T. Theoharis, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 640–649, 2007.

[29] G. Passalis, P. Perakis, T. Theoharis, and I. Kakadiaris, "Using facial symmetry to handle pose variations in real-world 3D face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1938–1951, 2011.

[30] P. Perakis, G. Passalis, T. Theoharis, and I. Kakadiaris, "3D facial landmark detection under large yaw and expression variations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1552–1564, 2013.

[31] Y. Lei, M. Bennamoun, M. Hayat, and Y. Guo, "An efficient 3D face recognition approach using local geometrical signatures," *Pattern Recognition*, vol. 47, no. 2, pp. 509–524, 2013.

[32] A. Mian, M. Bennamoun, and R. Owens, "An efficient multimodal 2D-3D hybrid approach to automatic face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1927–1943, 2007.

[33] K. I. Chang, K. Bowyer, and P. Flynn, "Adaptive rigid multi-region selection for handling expression variation in 3D face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 157, 2005.

[34] X. Li and F. Da, "Efficient 3D face recognition handling facial expression and hair occlusion," *Image and Vision Computing*, vol. 30, no. 9, pp. 668–679, 2012.

[35] Y. Ming and Q. Ruan, "Robust sparse bounding sphere for 3D face recognition," *Image and Vision Computing*, vol. 30, no. 8, pp. 524–534, 2012.

[36] Y. Wang, J. Liu, and X. Tang, "Robust 3D face recognition by local shape difference boosting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1858–1870, 2010.

[37] C. Xu, S. Li, T. Tan, and L. Quan, "Automatic 3D face recognition from depth and intensity gabor features," *Pattern Recognition*, vol. 42, no. 9, pp. 1895–1905, 2009.

[38] H. Mohammadzade and D. Hatzinakos, "Iterative closest normal point for 3D face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 381–397, 2013.

[39] M. Emambakhsh and A. Evans, "Self-dependent 3D face rotational alignment using the nose region," in *4th IET International Conference on Imaging for Crime Detection and Prevention (ICDP)*, pp. 1–6, 2011.

[40] A. Moorhouse, A. Evans, G. Atkinson, J. Sun, and M. Smith, "The nose on your face may not be so plain: Using the nose as a biometric," in *3rd IET International Conference on Crime Detection and Prevention (ICDP)*, pp. 1–6, 2009.

[41] Y. Lei, M., and A. A. El-Sallam, "An efficient 3D face recognition approach based on the fusion of novel local low-level features," *Pattern Recognition*, vol. 46, no. 1, pp. 24–37, 2013.

[42] H. Drira, B. B. Amor, A. Srivastava, M. Daoudi, and R. Slama, "3D face recognition under expressions, occlusions, and pose variations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2270–2283, 2013.

[43] R. Llonch, E. Kokiopoulou, I. Tosic, and P. Frossard, "3D face recognition using sparse spherical representations," in *19th IEEE International Conference on Pattern Recognition*, pp. 1–4, 2008.

[44] M. Segundo, L. Silva, O. Bellon, and C. Queirolo, "Automatic face segmentation and facial landmark detection in range images," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 40, no. 5, pp. 1319–1330, 2010.

[45] B. Horn, "Extended gaussian images," *Proceedings of the IEEE*, vol. 72, pp. 1671–1686, 1984.

[46] S. Malassiotis and M. Strintzis, "A three-dimensional fourier descriptor for human body representation/reconstruction from serial cross sections," *Computers and Biomedical Research*, vol. 20, pp. 125–140, 1987.

[47] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 433–449, 1999.

[48] M. Kazhdan, "An approximate and efficient method for optimal rotation alignment of 3D models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1221–1229, 2007.

[49] M. Kazhdan, , T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 156–164, 2003.

[50] S. Malassiotis and M. Strintzis, "Snapshots: A novel local surface descriptor and matching algorithm for robust 3D surface alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1285–1290, 2007.

[51] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607 –626, 2009.

[52] A. Batur and M. Hayes, "Adaptive active appearance models," *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1707–1721, 2005.

[53] X. Lu, D. Colbry, and A. Jain, "Matching 2.5D scans for face recognition," in *Biometric Authentication*, vol. 3072 of *Lecture Notes in Computer Science*, pp. 1–25, Springer Berlin / Heidelberg, 2004.

[54] T. Papatheodorou and D. Rueckert, "Evaluation of automatic 4D face recognition using surface and texture registration," in *6th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 321–326, 2004.

[55] T. Russ, M. Koch, and C. Little, "A 2D range Hausdorff approach for 3D face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 169, 2005.

[56] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, pp. 509–517, 1975.

[57] C. Queirolo, L. Silva, O. Bellon, and M. Segundo, "3D face recognition using simulated annealing and the surface interpenetration measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 206–219, 2010.

[58] N. Alyüz, B. Gökberk, and L. Akarun, "Regional registration for expression resistant 3-D face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 425–440, 2010.

[59] L. Silva, O. Bellon, and K. Boyer, "Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 762–776, 2005.

[60] S. Niyogi and W. Freeman, "Example-based head tracking," in *2nd IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 374 –378, 1996.

[61] D. Beymer, "Face recognition under varying pose," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 756–761, 1994.

[62] M. Emambakhsh, H. Ebrahimnezhad, and M. Sedaaghi, "Integrated region-based segmentation using color components and texture features with prior shape knowledge," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 4, pp. 711–726, 2010.

[63] P. Nair and A. Cavallaro, "3-D face detection, landmark localization, and registration using a point distribution model," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 611–623, 2009.

[64] D. Sim and R. Park, "Two-dimensional object alignment based on the robust oriented Hausdorff similarity measure," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 475–483, 2001.

[65] P. Yan and K. Bowyer, "A fast algorithm for ICP-based 3D shape biometrics," in *4th IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 213–218, 2005.

[66] A. Colombo, C. Cusano, and R. Schettini, "3D face detection using curvature analysis," *Pattern Recognition*, vol. 39, no. 3, pp. 444–455, 2006.

[67] K. Chang, W. Bowyer, and P. Flynn, "Multiple nose region matching for 3D face recognition under varying facial expression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1695–1700, 2006.

[68] P. Szeptycki, M. Ardabilian, and L. Chen, "A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking," in *3rd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6, 2009.

[69] C. Creusot, N. Pears, and J. Austin, "A machine-learning approach to keypoint detection and landmarking on 3D meshes," *International Journal of Computer Vision*, vol. 102, no. 1, pp. 146–179, 2013.

[70] A. Mian, M. Bennamoun, and R. Owens, "Keypoint detection and local feature matching for textured 3D face recognition," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 1–12, 2008.

[71] M. Segundo, L. Silva, O. Bellon, and C. Queirolo, "Automatic face segmentation and facial landmark detection in range images," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 40, no. 5, pp. 1319–1330, 2010.

[72] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1, pp. 43–72, 2005.

[73] C. Tomasi and T. Kanade, "Detection and tracking of point features," tech. rep., International Journal of Computer Vision, 1991.

[74] C. Harris and M. Stephens, "A combined corner and edge detector," in *In 4th Alvey Vision Conference*, pp. 147–151, 1988.

[75] D. Lowe, "Object recognition from local scale-invariant features," in *7th IEEE International Conference on Computer Vision*, pp. 1150–1157, 1999.

[76] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.

[77] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*, pp. 404–417, 2006.

[78] G. Zhang, X. Huang, S. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Advances in Biometric Person Authentication*, vol. 3338 of *Lecture Notes in Computer Science*, pp. 179–186, Springer Berlin Heidelberg, 2005.

[79] C. Podilchuk and X. Zhang, "Face recognition using DCT-based feature vectors," 1998. US Patent 5,802,208.

[80] H. Tanaka and M. Ikeda, "Curvature-based face surface recognition using spherical correlation-principal directions for curved object recognition," in *13th International Conference on Pattern Recognition (ICPR)*, pp. 638–642 vol.3, 1996.

[81] G. Gordon, "Face recognition based on depth and curvature features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 808–810, 1992.

[82] Y. Lee and J. Shim, "Curvature based human face recognition using depth weighted Hausdorff distance," in *IEEE International Conference on Image Processing (ICIP)*, pp. 1429–1432 Vol. 3, 2004.

[83] G. G. Gordon, "Face recognition based on depth maps and surface curvature," in *SPIE Geometric methods in Computer Vision*, pp. 234–247, 1991.

[84] V. Štruc and N. Pavešić, "The complete gabor-fisher classifier for robust face recognition," *EURASIP Advances in Signal Processing*, vol. 2010, p. 26, 2010.

[85] H. Li, D. Huang, J.-M. Morvan, L. Chen, and Y. Wang, "Expression-robust 3D face recognition via weighted sparse representation of multi-scale and multi-component local normal patterns," *Neurocomputing*, vol. 133, no. 0, pp. 179 – 193, 2014.

[86] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1063–1074, 2003.

[87] G. Pan, S. Han, Z. Wu, and Y. Wang, "3D face recognition using mapped depth images," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 175, 2005.

[88] L. Spreeuwers, "Fast and accurate 3D face recognition," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 389–414, 2011.

[89] Y. Wang, X. Tang, J. Liu, G. Pan, and R. Xiao, "3D face recognition by local shape difference boosting," in *European conference on Computer Vision (ECCV*, vol. 5302 of *Lecture Notes in Computer Science*, pp. 603–616, Springer Berlin Heidelberg, 2008.

[90] J. Lee and E. Milios, "Matching range images of human faces," in *3rd International Conference on Computer Vision (ICCV)*, pp. 722–726, 1990.

[91] N. Fisher, T. Lewis, and E. Embleton, *Statistical Analysis of Spherical Data*. Cambridge, UK: Cambridge Univ. Press, 1987.

[92] A. B. Moreno, Á. Sánchez, J. F. Vélez, and F. J. Díaz, "Face recognition using 3D surface-extracted descriptors," in *In Irish Machine Vision and Image Processing Conference (IMVIP)*, 2003.

[93] T. Nagamine, T. Uemura, and I. Masuda, "3D facial image analysis for human identification," in *11th International Conference on Pattern Recognition (ICPR)*, pp. 324–327, 1992.

[94] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Three-dimensional face recognition," *International Journal of Computer Vision*, vol. 64, pp. 5–30, 2005.

[95] H. Drira, , B. Amor, M. Daoudi, and A. Srivastava, "Nasal region contribution in 3D face biometrics using shape analysis framework," in *3rd International Conference on Advances in Biometrics*, pp. 357–366, 2009.

[96] J. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, pp. 1591–1595, 1996.

[97] S. Berretti, A. Del Bimbo, and P. Pala, "Sparse matching of salient facial curves for recognition of 3-D faces with missing parts," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 2, pp. 374–389, 2013.

[98] L. Ballihi, B. Ben Amor, M. Daoudi, A. Srivastava, and D. Aboutajdine, "Boosting 3-D-geometric features for efficient face recognition and gender classification," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1766–1779, 2012.

[99] S. Berretti, A. Del Bimbo, and P. Pala, "3D face recognition using isogeodesic stripes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2162–2177, 2010.

[100] C. Samir, A. Srivastava, and M. Daoudi, "Three-dimensional face recognition using shapes of facial curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1858–1863, 2006.

[101] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[102] Z. Sun, G. Bebis, and R. Miller, "Object detection using feature subset selection," *Pattern Recognition*, vol. 37, no. 11, pp. 2165–2176, 2004.

[103] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[104] B. Achermann, X. Jiang, and H. Bunke, "Face recognition using range images," in *International Conference on Virtual Systems and MultiMedia (VSMM)*, pp. 129–136, 1997.

[105] F. Tsalakanidou, D. Tzovaras, and M. G. Strintzis, "Use of depth and colour Eigenfaces for face recognition," *Pattern Recognition Letters*, vol. 24, pp. 1427–1435, 2003.

[106] A. Godil, S. Ressler, and P. Grother, "Face recognition using 3D facial shape and color map information: Comparison and combination," *SPIE*, vol. 5404, pp. 351–361, 2004.

[107] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[108] F. Al-Osaimi, M. Bennamoun, and A. Mian, "An expression deformation approach to non-rigid 3D face recognition," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 302–316, 2009.

[109] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[110] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *7th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 245–250, 2001.

[111] M. Bartlett, J. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1450–1464, 2002.

[112] B. A. Draper, K. Baek, M. S. Bartlett, and J. Beveridge, "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding*, vol. 91, no. 1, pp. 115–137, 2003.

[113] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[114] C. Hesher, A. Srivastava, and G. Erlebacher, "A novel technique for face recognition using range imaging," in *7th International Symposium on Signal Processing and Its Applications*, pp. 201–204 vol.2, 2003.

[115] J. Cook, V. Chandran, and S. Sridharan, "Multiscale representation for 3-D face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 529–536, 2007.

[116] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270 – 2285, 2005.

[117] K. Delac, M. Grgic, and S. Grgic, "Independent comparative study of PCA, ICA, and LDA on the FERET data set," *International Journal of Imaging Systems and Technology*, vol. 15, no. 5, pp. 252–260, 2005.

[118] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850–863, 1993.

[119] T. Russ, M. Koch, and C. Little, "3D facial recognition: a quantitative analysis," in *38th Annual 2004 International Carnahan Conference on Security Technology*, pp. 338 – 344, 2004.

[120] G. Pan, Z. Wu, and Y. Pan, "Automatic 3D face verification from range data," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 193–6 vol.3, 2003.

[121] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSbd: The extended M2VTS database," in *2nd Conference on Audio and Video-base Biometric Personal Verification (AVBPA99)*, 1999.

[122] F. Tsalakanidou, S. Malassiotis, and M. Strintzis, "Integration of 2D and 3D images for enhanced face authentication," in *6th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 266–271, 2004.

[123] F. Forster, P. Rummel, M. Lang, and B. Radig, "The HISCORE camera a real time three dimensional and color camera," in *International Conference on Image Processing (ICIP)*, pp. 598–601 vol.2, 2001.

[124] S. Lin, S. Kung, and L. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–132, 1999.

[125] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697–710, 2002.

[126] S. Lawrence, C. Giles, A. C. Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[127] Y. Lee, H. Song, U. Yang, H. Shin, and K. Sohn, "Local feature based 3D face recognition," in *Audio- and Video-Based Biometric Person Authentication*, vol. 3546, pp. 909–918, Springer Berlin / Heidelberg, 2005.

[128] M. Jones and P. Viola, "Face recognition using boosted local features," tech. rep., 2003. Mitsubishi Electric Research Laboratories.

[129] J. Lu, K. Plataniotis, A. Venetsanopoulos, and S. Li, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 166–178, 2006.

[130] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[131] W. Deng, J. Hu, and J. Guo, "Extended SRC: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–1870, 2012.

[132] L. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[133] M. Monwar and M. Gavrilova, "Multimodal biometric system using rank-level fusion approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 4, pp. 867–878, 2009.

[134] A. Kumar and S. Shekhar, "Personal identification using multibiometrics rank-level fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 5, pp. 743–752, 2011.

[135] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[136] M. Nielsen, "Neural networks and deep learning," tech. rep., 2014.

[137] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2559–2566, 2010.

[138] E. Humphrey, J. Bello, and Y. LeCun, "Feature learning and deep architectures: new directions for music informatics," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 461–481, 2013.

[139] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *Journal of Machine Learning Research*, vol. 11, p. 625660, 2010.

[140] H. Dibeklioğlu, B. Gökberk, and L. Akarun, "Nasal region-based 3D face recognition under pose and expression variations," in *3rd International Conference on Advances in Biometrics*, pp. 309–318, 2009.

[141] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[142] S. Bharadwaj, H. Bhatt, M. Vatsa, R. Singh, and A. Noore, "Quality assessment based denoising to improve face recognition performance," in *IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 140–145, 2011.

[143] W. Lin and M. Chen, "Automatic quality assessment and preprocessing for three-dimensional face recognition," in *International Conference on Information Security and Intelligence Control (ISIC)*, pp. 266–269, 2012.

[144] D. Colbry and G. Stockman, "Real-time identification using a canonical face depth map," *IET Computer Vision*, vol. 3, no. 2, pp. 74–92, 2009.

[145] A. Flint, A. Dick, and A. van den Hengel, "Local 3D structure recognition in range images," *IET Computer Vision*, vol. 2, no. 4, pp. 208–217(9), 2008.

[146] X. Sun, P. Rosin, R. Martin, and F. Langbein, "Noise in 3D laser range scanner data," in *IEEE International Conference on Shape Modeling and Applications*, pp. 37–45, 2008.

[147] V. Struc, "The PhD face recognition toolbox," February 2012. http://www.mathworks.co.uk/matlabcentral/fileexchange/35106-the-phd-face-recognition-toolbox, access time: 13-11-2014.

[148] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[149] M. Emambakhsh, M. Sedaaghi, and H. Ebrahimnezhad, "Locating texture boundaries using a fast unsupervised approach based on clustering algorithms fusion and level set," in *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 129–134, 2009.

[150] F. D'Almeida, "Nonlinear diffusion toolbox," July 2003. http://www.mathworks.co.uk/matlabcentral/fileexchange/3710-nonlinear-diffusion-toolbox, access time: 19-Oct-2014.

[151] R. Raguram, C. Wu, J.-M. Frahm, and S. Lazebnik, "Modeling and recognition of landmark image collections using iconic scene graphs," *International Journal of Computer Vision*, vol. 95, no. 3, pp. 213–239, 2011.

[152] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[153] G. Dalley and P. Flynn, "Range image registration: A software platform and empirical evaluation," in *3rd International Conference on 3-D Digital Imaging and Modeling*, pp. 246–253, 2001.

[154] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[155] G. Slabaugh, "Computing euler angles from a rotation matrix," tech. rep.

[156] J. Weston and C. Watkins, "Multi-class support vector machines," in *Proceedings of European Symposium on Artificial Neural Networks (ESANN)*, 1999.

[157] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[158] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[159] D. Smeets, J. Keustermans, D. Vandermeulen, and P. Suetens, "meshsift: Local surface features for 3D face recognition under expression variations and partial data," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 158 – 169, 2013.

[160] H. Li, D. Huang, P. Lemaire, J.-M. Morvan, and L. Chen, "Expression robust 3D face recognition via mesh-based histograms of multiple order surface differential quantities,"

in *18th IEEE International Conference on Image Processing (ICIP)*, pp. 3053–3056, 2011.

[161] N. Alyüz, B. Gökberk, and L. Akarun, "A 3D face recognition system for expression and occlusion invariance," in *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–7, 2008.

[162] X. Liu, S. Shan, and X. Chen, "Face recognition after plastic surgery: A comprehensive study," in *11th Asian Conference on Computer Vision (ACCV)*, vol. 7725 of *Lecture Notes in Computer Science*, pp. 565–576, Springer Berlin Heidelberg, 2013.